



Domòtica aplicada a un local

Oscar Solans Caballer

Grau Tecnologies de la Telecomunicació

Arduino

Antoni Morell Pérez

Pere Tuset Peiró

Gener 2019



Aquesta obra està subjecta a una llicència de Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Domòtica aplicada a un local</i>
Nom de l'autor:	<i>Oscar Solans Caballer</i>
Nom del consultor/a:	<i>Antoni Morell Pérez</i>
Nom del PRA:	<i>Pere Tuset Peiró</i>
Data de lliurament (mm/aaaa):	<i>01/2019</i>
Titulació o programa:	<i>Grau de Tecnologies de la Telecomunicació</i>
Àrea del Treball Final:	<i>Arduino</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>Raspberry Pi – MQTT - OpenHAB</i>
Resum del Treball (màxim 250 paraules): <i>Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball</i>	
<p>Aquest treball pretén implementar un sistema domòtic completament funcional en un local comercial. Aquest escenari no queda definit en un entorn físic si no que pretén ser adaptable a qualsevol espai, per tant, s'utilitza una maqueta a mode de simulació del sistema.</p> <p>Per a aconseguir implementar el projecte, s'utilitza una placa controladora Arduino, la qual rebrà la informació dels sensors i enviarà ordres als diferents actuadors. Aquest dispositiu, es comunica amb una placa computadora, anomenada Raspberry Pi.</p> <p>Aquesta última, queda definida llavors com a node central del sistema, ja que és l'encarregada de gestionar la informació rebuda i prendre decisions d'actuació en base a la configuració definida en el sistema.</p> <p>La comunicació entre aquests dos nodes, es realitza a través del protocol MQTT, configurant l'Arduino com a client i la Raspberry com a servidor. En concret, el servidor triat s'anomena Mosquitto.</p> <p>Amb la finalitat d'establir una interfície de treball suportada tant pel protocol de comunicació MQTT com pel node central del sistema, s'utilitza la plataforma OpenHAB, la qual permet visualitzar la informació proporcionada pel sistema i actuar sobre ell.</p> <p>A més a més, i per tal de poder actuar telemàticament sobre la maqueta, es crea una VPN i s'instal·la una càmera com a dispositiu per a control visual del sistema.</p> <p>Finalment, s'observa com la metodologia seguida ha permès obtenir els resultats esperats, assolint així els objectius definits a l'inici del document.</p>	

Abstract (in English, 250 words or less):

The aim of this project is to implement a completely functional domotic system in a commercial place. The scenario is not defined in a physical environment, but instead it pretends to be adaptable to any space and therefore, a model is used as a simulation of the system.

In order to implement the project, an Arduino controller board is used, which will receive information from the sensors and send orders to the different actuators. This device communicates with a computer board, called Raspberry Pi.

Raspberry Pi is then defined as the central node of the system, as it is responsible for managing the information received and taking action decisions based on the configuration defined in the system.

The communication between these two nodes is carried out through the MQTT protocol, configuring Arduino as a client and Raspberry as a server. In particular, the chosen server is called Mosquitto.

In order to establish a work interface supported by the MQTT communication protocol and by the central node of the system, the OpenHAB platform is used, which allows to visualize the information provided by the system and to act on it.

In addition, and in order to be able to act telematically on the model, a VPN is created and a camera is installed as a device for the visual control of the system.

Finally, it can be seen that the methodology followed has allowed to obtain the expected results, achieving the objectives defined at the beginning of the document.

Índex de continguts

1. Descripció del projecte	3
2. Objectius del projecte.....	5
3. Viabilitat.....	6
4. Planificació del treball.....	8
5. Estat de l'art.....	10
6. Arquitectura i disseny del sistema.....	12
6.1 Arquitectura i funcionament bàsic.....	12
6.2 Descripció de tecnologies utilitzades	13
6.2.1 Arduino Mega	13
6.2.2 Sensors.....	15
6.2.3 Actuadors.....	19
6.2.4 Raspberry Pi 3B	21
6.2.5 OpenVPN.....	22
6.2.6 Protocol MQTT	22
6.2.7 OpenHAB.....	24
6.3 Configuració de xarxa	25
7. Implementació.....	26
7.1 Configuració Raspberry.....	26
7.2 Configuració del sistema OpenHAB2	27
7.2.1 Definició d'ítems	29
7.3 Instal·lació i configuració Arduino	30
7.3.1 Estructura del codi.....	30
7.3.2 Arduino i Ethernet Shield WS5100	32
7.3.3 Arduino i MQTT	32
7.4 Maqueta i interconnexió de sensors i actuadors	33
7.4.1 Interconnexió de sensors i actuadors	33
7.5 Definició regles OpenHAB	38
8. Comprovació i resultats	40
8.1 Comprovació del sistema	40
8.2 Resultats i conclusions.....	45
Bibliografia	47

ANNEX I - Instal·lació Raspberry	49
ANNEX II - Instal·lació i configuració PiVPN.....	50
ANNEX III - Instal·lació OpenHAB.....	54
ANNEX IV - Instal·lació broker MQTT: Mosquitto	58
ANNEX V - Instal·lació servidor Samba	60
ANNEX VI - Instal·lació addicional	62
VI.I - OpenVPN-GUI	62
VI.II - Notepad++	62
VI.III - VNC Viewer	63
VI.IV - NoIP	64
ANNEX VII - Instal·lació Camera Suite per a GoPRO Hero4	65

Índex de il·lustracions

Il·lustració 1. Esquema funcionament del sistema.....	12
Il·lustració 2. Arduino MEGA	13
Il·lustració 3. Arduino Ethernet Shield	14
Il·lustració 5. Arduino Power Supply	15
Il·lustració 6. Funcionament HC-SR04.....	15
Il·lustració 7. HC-SR04	15
Il·lustració 8. DHT22	16
Il·lustració 9. PIR.....	16
Il·lustració 10. Funcionament PIR	17
Il·lustració 11. Water Sensor	17
Il·lustració 12. ACS712	17
Il·lustració 13. LDR55	18
Il·lustració 14. Esquema muntatge LDR55	18
Il·lustració 15. MQ-7.....	18
Il·lustració 16. Funcionament servomotor	19
Il·lustració 17. Mòdul relés.....	19
Il·lustració 18. LED.....	20
Il·lustració 19. Parts buzzer	20
Il·lustració 20. Raspberry Pi 3B	21
Il·lustració 21. Funcionament protocol MQTT	23
Il·lustració 22. Exemple interfície OpenHAB	25

Il·lustració 23. Configuració DHCP	25
Il·lustració 24. Activar VNC i SSH a Raspberry	26
Il·lustració 25. Configuració xarxa Raspberry	26
Il·lustració 26. Comprovació configuració de xarxa	27
Il·lustració 27. Edició fitxer configuració mosquittot	27
Il·lustració 28. App mòbil OpenHAB	29
Il·lustració 29. BasicUI OpenHAB.....	30
Il·lustració 30. HABPanel OpenHAB	30
Il·lustració 31. Maqueta projecte	33

Índex de taules

Taula 1. Característiques Arduino MEGA	14
Taula 2. Característiques DHT22	16
Taula 3. Característiques Raspberry Pi 3B	22
Taula 4. Descripció elements OpenHAB.....	29
Taula 5. Descripció de blocs sketch Arduino	31
Taula 6. Esquema circuital dels diferents elements.....	37
Taula 7. Resultats del projecte	45

1.Descripció del projecte

Aquest projecte té l'objectiu d'implementar un sistema domòtic en un local comercial, el qual estarà connectat a la xarxa local i serà accessible i controlable mitjançant una aplicació.

Pel que fa a les tecnologies que s'utilitzaran, aquestes són:

- Arduino: controlador basat en hardware i software *opensource* que permet connectar perifèrics (sensors i actuadors en el nostre cas) a les diferents entrades i sortides d'un microcontrolador.
- Arduino Ethernet Shield: permet connectar el nostre dispositiu Arduino a la xarxa.
- Raspberry Pi 3: computador de mides reduïdes i baix cost que rebrà la informació facilitada per Arduino i, en base a unes regles que definirem, li enviarà ordres perquè aquestes siguin executades.
- Protocol MQTT (*Message Queue Telemetry Transport*) [16]: utilitzat en la comunicació M2M (*machine-to-machine*) en el Internet de les Coses (IoT). Permet comunicar el dispositiu Arduino amb la Raspberry Pi de forma bidireccional mitjançant el protocol TCP/IP.
- OpenHAB: software *opensource* que permet integrar diferents sistemes de domòtica i automatitzar diverses funcions en base a unes regles definides per l'usuari. A més, ofereix un entorn visual atractiu que pot ser instal·lat en qualsevol dispositiu que suporti Java, com per exemple un terminal mòbil o la mateixa Raspberry Pi, per al control per part de l'usuari.

A banda de tot aquest maquinari i software, s'utilitzarà un seguit de sensors connectats a l'Arduino per a capturar la informació que interressi del medi i enviar-la al dispositiu Raspberry Pi:

- Sensor de temperatura i humitat.
- Sensor ultrasònic.
- Detector de moviment.
- Detector de gasos i foc.
- Detector d'inundació.
- Sensor de corrent.
- Sensor de lluminositat.

I per actuar en base a les regles definides a OpenHAB, tot un seguit d'actuadors també connectats a l'Arduino:

- Servomotor.

- Relé.
- Led.
- *Buzzer.*

2.Objectius del projecte

Es defineixen els següents objectius:

- Implementar un sistema domòtic que es comuniqui sota el protocol MQTT amb missatges que requereixen poc ample de banda.
- Utilitzar l'aplicació opensource OpenHAB per interactuar amb el sistema implementat.
- Gestionar una xarxa VPN per accedir al sistema amb independència del sistema natiu d'OpenHAB.
- Utilitzar interrupcions d'Arduino en cas de detecció d'inundació o foc, per lliurar l'Arduino de temps de computació i estalviar energia.
- Configurar les alarmes i la seva conseqüent notificació al dispositiu mòbil.
- Controlar remotament l'apertura i tancament de la porta i l'encesa i apagada d'elements elèctrics.

3.Viabilitat

En l'actualitat, gairebé la totalitat de les llars i locals disposen de connexió a la xarxa i dispositius mòbils amb connexió de dades inclosa, cosa que facilita la viabilitat de l'aplicació del projecte a la realitat.

A més, les tecnologies utilitzades tenen un valor econòmic assequible per gairebé la totalitat de la població:

ELEMENT	PREU
Arduino Mega + Power Suply	20.76 €
Arduino Ethernet Shield	6.23 €
Raspberry Pi 3	27.12 €
Sensors	26.32 €
Actuadors	21.56 €
Cables i altres elements	32.45 €
Total	134.44 €

A aquest valor, cal sumar-li la despesa en termes de temps de dedicació al projecte en sí:

ETAPA	DURADA	PREU (20 €/h)
Planificar	16 h	320 €
Definir arquitectura i disseny del sistema	40 h	800 €
Implementació tècnica	160 h	3200 €
Comprovació de resultats	16 h	320 €
Total		4640 €

Així doncs, el cost total del projecte seria de 4774.44€.

Per tal de fer viable el projecte, es planteja una venda del sistema complet a 1000€/estància.

A més, es fa extensible l'ús d'aquests sistema per a qualsevol tipus d'habitació d'una llar,

adaptant-se en cada cas a les necessitats pel que fa a sensors i actuadors, ja que el disseny presenta una gran escalabilitat i moltes possibles configuracions tot realitzant petits canvis de programació.

Per tal d'assolir un benefici del 20% amb el nostre projecte, es conclou que aquest seria viable a partir de la venda de 6 unitats del sistema, assolint un preu de venda de 6000€.

Per tant, pensant en la possibilitat d'instal·lar-ho a diferents locals i a habitatges que disposin de 3 o 4 estàncies, el projecte presentarà beneficis amb poques vendes a diferents clients que superaran amb escreix el cost mencionat en un curt termini de temps.

En cas de voler fer atractiu el projecte per a un inversor, es planteja la possibilitat de presentar un projecte comú per a tots els locals d'un determinat centre comercial:

Centre Comercial	Núm. locals	Cost	Venta	Benefici
Glòries	126	584.640 €	756.000 €	171.360 €
La Maquinista	242	1.122.880 €	1.452.000 €	329.120 €

Ara bé, els beneficis són una aproximació a la realitat, ja que aquests disminuirien a l'augmentar el nombre de recursos humans destinats al projecte, cosa que suposa un increment en l'element de preu més elevat del projecte.

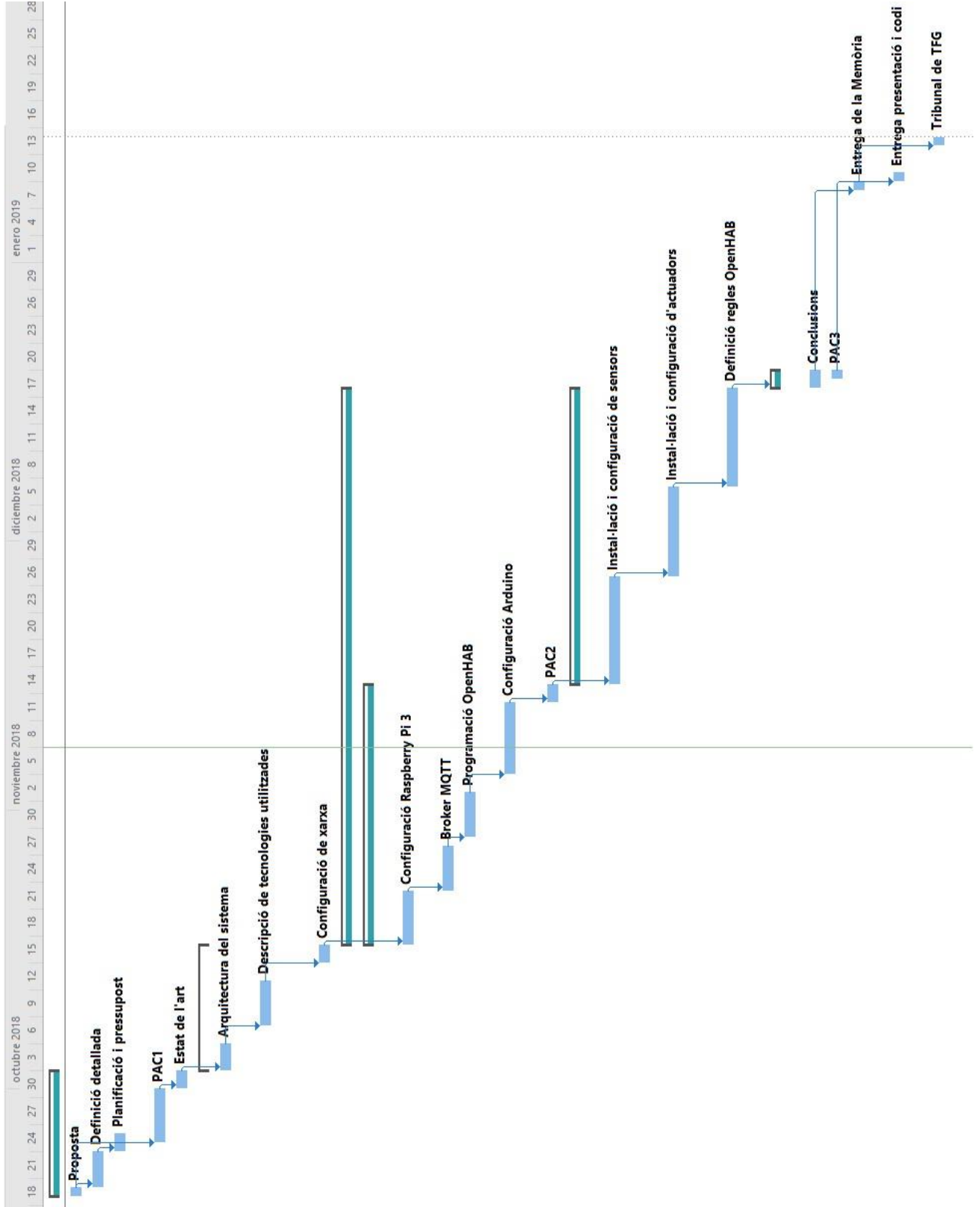
Pensant en un equip format per 10 persones (és a dir, 9 treballadors/es addicionals), els beneficis reals aproximats serien:

Centre Comercial	Núm. locals	Benefici	Cost adicional (material + RRHH)	Benefici real
Glòries	126	171.360 €	58.699,44 €	112.660.56 €
La Maquinista	242	329.120 €	74.294,48 €	254.825,52 €

4. Planificació del treball

A continuació es mostren les diferents tasques a realitzar i el corresponent diagrama de Gant:

Nombre tasca	Duració	Inici	Fi
Descripció del projecte	12 días	mié 9/19/18	mar 10/2/18
Proposta	1 día	mié 9/19/18	mié 9/19/18
Definició detallada	3 días	jue 9/20/18	dom 9/23/18
Planificació i pressupost	2 días	lun 9/24/18	mar 9/25/18
PAC1	5 días	mar 9/25/18	dom 9/30/18
Estat de l'art	2 días	lun 10/1/18	mar 10/2/18
Arquitectura i disseny	10 días	mié 10/3/18	mar 10/16/18
Arquitectura del sistema	3 días	mié 10/3/18	vie 10/5/18
Descripció de tecnologies utilitzades	5 días	lun 10/8/18	vie 10/12/18
Configuració de xarxa	2 días	lun 10/15/18	mar 10/16/18
Implementació tècnica	45 días	mié 10/17/18	lun 12/17/18
Connexió entre maquinari	22 días	mié 10/17/18	mié 11/14/18
Configuració Raspberry Pi 3	4 días	mié 10/17/18	lun 10/22/18
Broker MQTT	5 días	mar 10/23/18	sáb 10/27/18
Programació OpenHAB	5 días	lun 10/29/18	vie 11/2/18
Configuració Arduino	6 días	lun 11/5/18	lun 11/12/18
PAC2	2 días	mar 11/13/18	mié 11/14/18
Programació sensors i actuadors	23 días	jue 11/15/18	lun 12/17/18
Instal·lació i configuració de sensors	8 días	jue 11/15/18	lun 11/26/18
Instal·lació i configuració d'actuadors	8 días	mar 11/27/18	jue 12/6/18
Definició regles OpenHAB	7 días	vie 12/7/18	lun 12/17/18
Proves finals i Resultats	2 días	mar 12/18/18	mié 12/19/18
Conclusions	2 días	mar 12/18/18	mié 12/19/18
PAC3	1 día?	mié 12/19/18	mié 12/19/18
Entrega de la Memòria	1 día?	mié 1/9/19	mié 1/9/19
Entrega presentació i codi	1 día	jue 1/10/19	jue 1/10/19
Tribunal de TFG	1 día	lun 1/14/19	lun 1/14/19



5. Estat de l'art

L'inici de la domòtica té lloc a la dècada dels 70 amb el sistema de protocol estàndard X10 [13]. Gràcies a la seva senzillesa i gran accessibilitat, van aparèixer un gran nombre d'aplicacions software i hardware que van ser instal·lades de forma massiva a produccions immobiliàries principalment als Estat Units.



El principal problema d'aquests sistema és que es basa en la transmissió de senyal sobre corrents portadores, cosa que fa que la qualitat dels senyals sigui força vulnerable a les alteracions que aquestes portadores puguin patir [14]. Amb aquesta característica, es descartava la possibilitat de controlar sistemes complexos i es centraven a funcions senzilles per a sistemes amb 2 estats.

Si bé als anys 80 es van deixar de fer proves pilot i es va estendre l'ús comercial d'aquests sistemes, no va ser fins a finals dels 90 quan va aparèixer l'estàndard KNX, amb el qual es pretenia donar més utilitats a automats programables dins les llars, ja que fins aleshores es centraven en tasques de regulació i control a fàbriques. Paral·lelament, l'any 1999 naixia el protocol de comunicació LonTalk de l'empresa LonWorks, el qual va ser normalitzat com a estàndard de control de xarxes [15].



Ambdues tecnologies, que deixaven de banda la xarxa elèctrica i basaven la comunicació en la xarxa TCP/IP, van ser fortament acceptades per les empreses del sector elèctric, de manera que van créixer de manera important. Tant és així, que el sector va veure un fort potencial en aquestes tecnologies i van aparèixer els primers protocols propietaris, tant de gama mitja com de gama alta, de diferents empreses.

Pel que fa a les llars, que demandaven tan sols parts d'aquests sistemes domòtics (com per exemple, alarmes), van començar a aparèixer els kits domòtics que, treballant de manera aïllada, permeten controlar tot un seguit d'entrades i sortides amb alguna funció extra de control telemàtic.



A partir dels 2000, concretament al 2006, sorgeixen els primers sistemes domòtics wireless RF, els quals utilitzen els protocols Zigbee (Control4), capaç d'assolir llargues distàncies de transmissió (100m), i Z-Wave, per a distàncies inferiors a 30

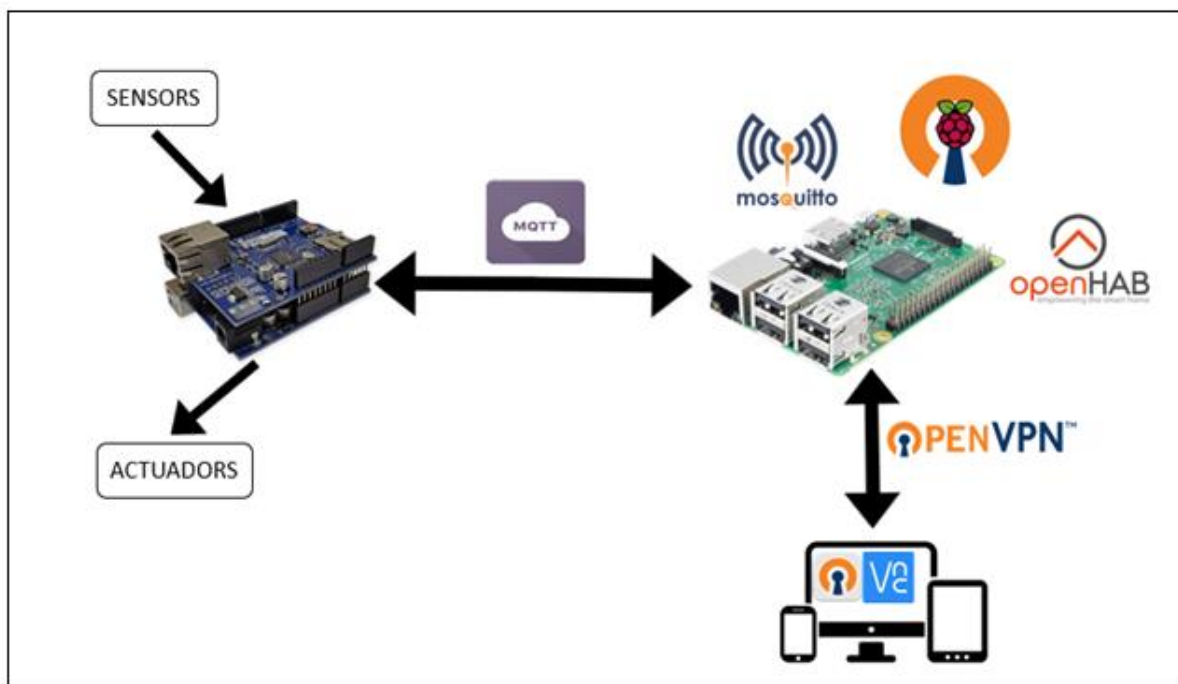
metres. Amb aquestes tecnologies, les quals treballen a freqüència 2.4GHz i 1GHz respectivament, s'amplia el ventall de possibilitats i es dona solució a aquelles funcions que els sistemes cablejat no són capaces de satisfer (com per exemple, comandaments de televisor per radiofreqüència).



6.Arquitectura i disseny del sistema

6.1 Arquitectura i funcionament bàsic

L'arquitectura del projecte serà la següent:



Il·lustració 1. Esquema funcionament del sistema

Tal i com s'observa en el gràfic, el node principal és la Raspberry Pi, on s'instal·len un 'broker' MQTT i la interfície OpenHAB per a controlar la domòtica, i el node secundari serà el dispositiu Arduino Mega.

Ambdós dispositius, Arduino i Raspberry, disposaran d'una adreça concreta de la xarxa local dins el rang d'adreçament estàtic de la mateixa per establir la comunicació sota el protocol TCP/IP.

Dins del protocol de comunicació M2M MQTT, el primer d'ells actuarà com a 'client' mentre que el segon actuarà com a 'servidor' i es comunicaran a través d'un 'broker', encarregat de gestionar la xarxa i transmetre els diferents missatges.

Pel que fa la interacció amb el sistema, es definiran unes regles i es programarà la interfície a OpenHAB per visualitzar i facilitar el control del sistema. Aquest també actuarà com a client del protocol MQTT.

Com a sistema d'accés addicional al sistema i per no dependre d'OpenHAB en cas de fallida del sistema i per treballar remotament sobre l'Arduino connectat a la xarxa local, es crea una VPN. Per tal d'assolir aquest objectiu, s'instal·la un servidor d'OpenVPN a

Raspberry (PiVPN) i un client OpenVPN GUI a l'ordinador des d'on es realitzarà el projecte, per tal d'accedir a la xarxa local des de qualsevol lloc, sense necessitat de tenir accés WiFi o Ethernet a la xarxa local.

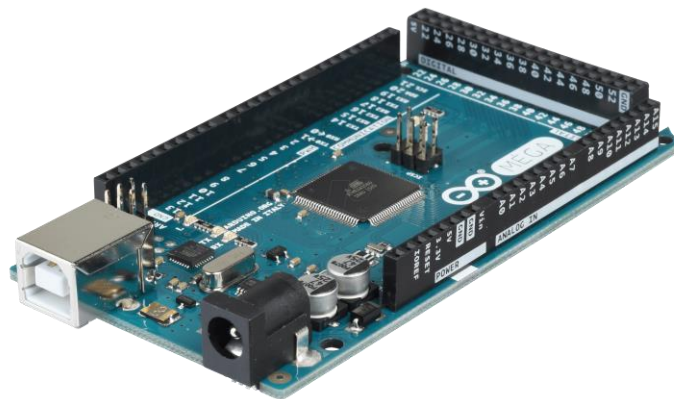
Per últim, s'instal·la al mateix ordinador de control, el software VNC Viewer per tal d'accedir a la interfície gràfica de la Raspberry i poder treballar sobre ella, ja que es on es mantindrà connectat l'Arduino per USB per tenir-hi accés remot i carregar-hi programes a través del seu propi IDE.

6.2 Descripció de tecnologies utilitzades

6.2.1 Arduino Mega

La família Arduino està composta per una gran varietat de hardware i software opensource.

En concret, per aquest projecte s'utilitzarà la placa Arduino Mega, construïda amb un microcontrolador Atmega2560 i programada en un entorn de desenvolupament amb llenguatge Processing/Wiring a través del port serial (mitjançant USB).



Il·lustració 2. Arduino MEGA

Aquest Arduino concret, presenta molta més capacitat de processament que la resta de les plaques de la família (Nano, UNO...), cosa que el fa especialment adient per a controlar un gran nombre de sensors i actuadors i comunicar-se bidireccionalment amb el dispositiu Raspberry a través del protocol MQTT, com és el cas.

En la taula s'observen les principals característiques del dispositiu:

MICROCONTROLLER	ATMEGA2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V

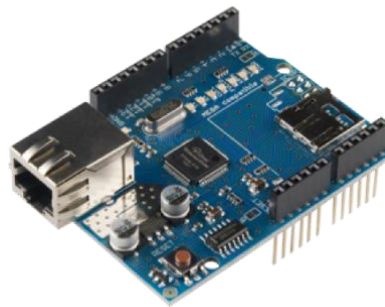
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13

Taula 1. Característiques Arduino MEGA

6.2.1.1 Complements Arduino Mega

En concret, s'utilitzen 2 complements pel que fa a l'alimentació de la placa i a la seva connexió a internet:

- Ethernet Shield: utilitzar per a connectar el dispositiu Arduino a la xarxa, ja que aquest no disposa de port Ethernet o antena WiFi. Així doncs, el dispositiu triat és el W5100.



Il·lustració 3. Arduino Ethernet Shield

- Breadboard Power Supply: aquest dispositiu s'encarrega d'alimentar Arduino amb els 7-12V necessaris que proporciona un adaptador de corrent i alhora alimentarà una protoboard amb 3.3V o 5V, segons ens interressi.



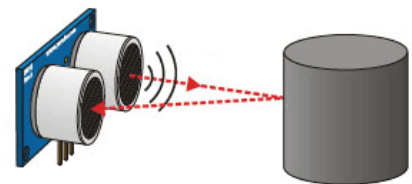
Il·lustració 4. Arduino Power Supply

6.2.2 Sensors

Es detallen a continuació els diferents sensors utilitzat per la realització del projecte.

6.2.2.1 Sensor d'ultrasons: HC-SR04 [5]

Aquest sensor és capaç de mesurar distàncies mitjançant l'enviament de polsos d'alta freqüència. Un cop llançat el pols, aquest rebota en l'objecte en qüestió i triga un determinat temps a tornar al sensor, el qual capta el pols rebotat mitjançant un micròfon adequat per a treballat a aquesta freqüència. Acte seguit, i mitjançant una fórmula matemàtica, ohm pot mesurar la distància amb una simple conversió del temps que triga el pols a tornar a la distància recorreguda pel mateix:



Il·lustració 5. Funcionament HC-SR04

$$Distància (cm) = \frac{Temps(\mu s)}{29.2 \cdot 2}$$



Il·lustració 6. HC-SR04

Pel que fa al rang de mesura pel cas del HC-SR04, aquest s'estableix en el marge de 2cm a 4m, amb una resolució de 0.3cm, tot i que a la pràctica s'estableix un rang de 20cm als 2m per garantir una certa precisió en la mesura.

Algun dels inconvenients que presenten aquests sensors, són la baixa precisió, el funcionament a zones exteriors i aire lliure, el mal funcionament en cas de múltiples objectes en el tram de mesura i la necessitat d'orientar ben bé l'objecte del qual se'n vol obtenir la distància, ja que aquesta orientació pot fer que s'obtinguin resultats falsejats (per exemple, en cas d'inclinació de la superfície de rebot del pols).

6.2.2.2 Sensor de temperatura i humitat: DHT22

En aquest projecte s'utilitza el sensor DHT22 [3], capaç d'obtenir alhora mesures de temperatura i humitat mitjançant un processador intern que genera un senyal digital capaç de ser llegida per Arduino.



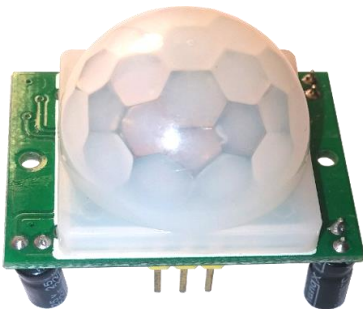
Il·lustració 7. DHT22

Les principals característiques, que permeten l'ús del DHT 22 per projectes reals de monitorització amb una precisió acceptable, s'especifiquen a la taula següent:

Característica	Valor
Mesura d'humitat	del 0 al 100% amb una precisió del 2 al 5%
Freqüència de mostreig	2Hz
Mesura de temperatura	de -40 a 125 °C amb precisió de 0.5 °C

Taula 2. Característiques DHT22

6.2.2.3 Sensor de moviment (PIR): HC-SR501 [4]

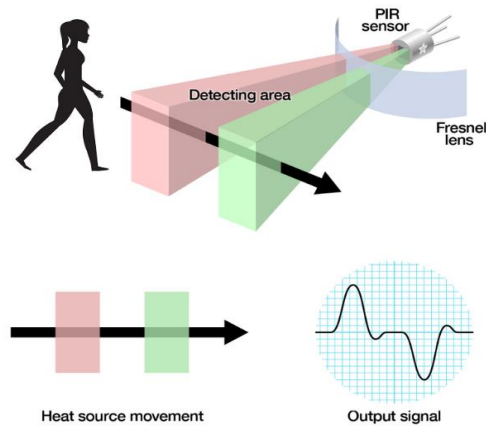


Il·lustració 8. PIR

Els sensors PIR (Passive Infrared Sensor) són dispositius que contenen un piezoelèctric capaç de detectar el moviment basant-se en la mesura de la radiació infraroja, ja que tots els elements i éssers vius emeten una certa quantitat d'aquesta energia, i posteriorment convertir aquesta captació en un senyal elèctric.

Cada sensor PIR està dividit internament en dos camps que, en cas de captar la mateixa quantitat d'infrarojos, produeixen un senyal elèctric nul i en cas contrari, es genera un senyal elèctric diferencial. Així doncs, si un objecte travessa un dels dos camps, s'emet un senyal digital cap a Arduino, en el nostre cas.

Un altre element a destacar d'aquest sensor és la seva òptica, composta per una cúpula formada per diferents lents de Fresnel, que divideixen l'espai per zones i s'encarreguen de dirigir la radiació d'infrarojos cap a cadascun dels camps del sensor, permeten així que quan una de les lents capti una radiació diferents, es generi el senyal diferencial mencionat anteriorment.



Il·lustració 90. Funcionament PIR

A més a més, aquest sensor permet ajustar la sensibilitat per tal d'evitar falsos positius produïts per petites variacions, per exemple, de lluminositat.

6.2.2.4 Sensor d'inundació [8]

Aquest tipus de sensors disposen d'unes tires conductores les quals no fan contacte entre elles. En entrar en contacte amb l'aigua, al ser aquest un material conductor, es genera connexió entre elles i varien la conductivitat del sensor. Aquest fet produeix un senyal digital que s'envia a Arduino.

El valor del senyal digital pot prendre valor baix (o 0) en cas de que no detecti aigua, o valor alt (1) en cas de detecció d'aigua sobre les seves tires conductores.



Il·lustració 10. Water Sensor

6.2.2.5 Sensor de corrent: ACS712 [9][10]



Il·lustració 11. ACS712

El sensor ACS712 és capaç de detectar corrent contínua i corrent alterna i mesurar la intensitat elèctrica que el travessa. Així doncs, podem mesurar a més, la potència que una càrrega endollada a ell consumeix.

El corrent flueix per un canal de coure intern, on es genera un camp magnètic que un sensor hall de precisió i baix offset detecta i converteix a una tensió proporcional al corrent que travessa el dispositiu.

Tot i que el sensor es calibra a fabricació, cal sempre un petit ajust per a que cada sensor concret funcioni de manera fiable. Aquest procés, es realitza mitjançant un filtratge de les lectures del lector basat en una mitjana aritmètica d'un determinat nombre de lectures

consecutives i a través d'un procés de calibratge de la sensibilitat del mateix a través de la mesura obtinguda per a dos corrent coneguts, a través de la fórmula:

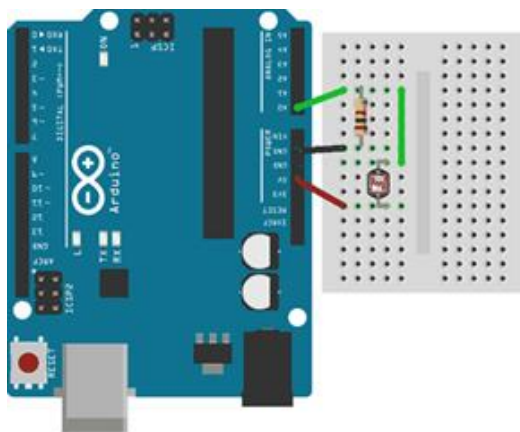
$$Sensibilitat = \frac{V_2 - V_1}{I_2 - I_1}$$

6.2.2.6 Sensor de Iluminositat: LDR55

Aquest sensor és un fotoresistor que varia la seva resistència en variar el nivell de llum que rep [2]. En concret, quanta més llum, menys resistència. Tot i això, cal tenir en compte que aquesta variació és lenta (de 20 a 100ms), i per tant no captarà variacions ràpides, cosa que farà per altra banda que presenti una gran estabilitat.



Il·lustració 12. LDR55



Si connectem el sensor a una de les entrades analògiques d'Arduino a través d'una resistència de 10kΩ, podrem obtenir una estimació del nivell en tant per cent de llum en funció de la resistència que presenti en un determinat moment.

Il·lustració 13. Esquema muntatge LDR55

6.2.2.7 Sensor detector de incendis: MQ-7 [7]

Aquest tipus de dispositius de la família MQ estan formats per un sensor electró-químic que varia la seva resistència al entrar en contacte amb diferents substàncies. Per tal que la lectura pugui ser fiable, aquests disposen d'un escalfador intern, el qual ha d'arribar a la seva pròpia temperatura de funcionament.

En aquest projecte, i amb la finalitat de detectar possibles incendis, es tria el sensor MQ-7, capaç de detectar CO₂ i incendis.



Il·lustració 14. MQ-7

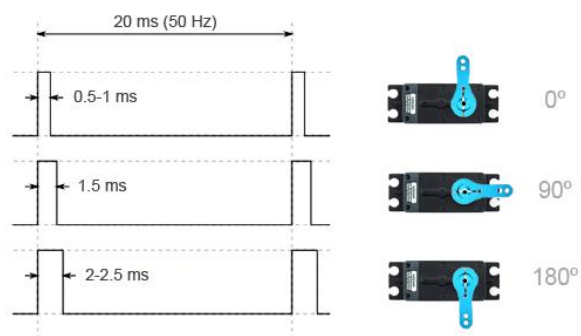
6.2.3 Actuadors

Es detallen a continuació els diferents sensors utilitzat per la realització del projecte.

6.2.3.1 Servomotor

A diferència d'altres tipus de motors on el que controlem és la velocitat de gir, en els servomotors indiquem l'angle desitjat on volem que aquest es posicioni dins del marge 0°-180°.

Per tal de situar el servo en la posició desitjada, es transmet un pols de senyal amb període de 20ms, i l'amplada d'aquest ens determina la posició tal i com s'observa a la imatge:



Il·lustració 15. Funcionament servomotor

En el seu interior, hi consta un mecanisme reductor que proporciona un alt par i una gran precisió, però per contra, la velocitat de gir és petita.

6.2.3.2 Relés [11][12]

Un relé permet a Arduino controlar càrregues d'un nivell de tensió/intensitat molt superiors a les que l'electrònica pot suportar. En el projecte s'utilitzen relés que permeten controlar càrregues de corrent altern de 220V amb intensitats de 10A.

Pel que fa al seu comportament, equival al d'un interruptor convencional o al d'un commutador, ja que és capaç de tancar o obrir un circuit, això sí, de forma electrònica. El temps de commutació d'un relé és d'un 10ms.



Il·lustració 16. Mòdul relés

Un relé està format per 2 circuits:

- Primari: part del relé que es connecta a Arduino i rep el senyal d'encès i apagat.

- Secundari: interruptor que encén o apaga la càrrega.

Els relés disposen normalment de 2 contactes al circuit secundari:

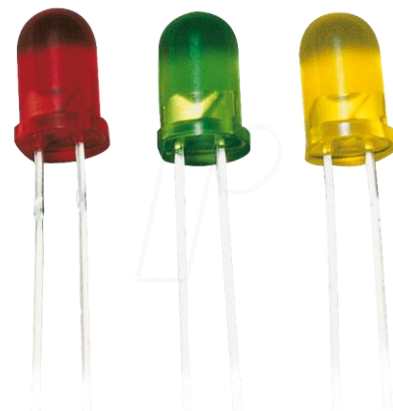
- Comú (C).
- Normalment Obert (NO): terminal aïllat de C quan el relé està apagat.
- Normalment Tancat (NC): terminal aïllat de C quan el relé està encès.

Així doncs, si desitgem utilitzar-lo com a interruptor, connectarem la carrega a C i a un dels altres dos indiferentment. En cas d'utilitzar-lo com a commutador, s'utilitzarien les 3 entrades de les que disposa aquesta banda del relé.

6.2.3.3 LED

Un LED és un simple díode que emet llum al ser travessat per un corrent elèctric quan aquest es troba polaritzat de manera directa. Aquesta llum, a més, es proporcional a aquest corrent.

Així doncs, al ser dispositius que tan sols permeten el pas d'un corrent en un determinat sentit, cal connectar l'ànode al voltatge positiu i el càtode al negatiu.

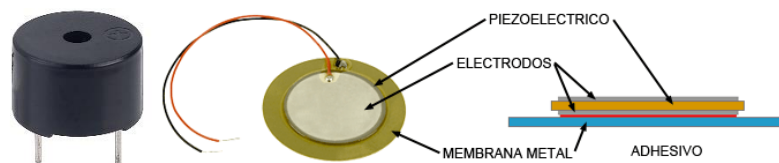


Il·lustració 17. LED

Una pràctica habitual, consisteix a connectar-lo en sèrie amb una resistència de valor adequat per tal d'evitar possibles danys en el LED. Aquest valor se situa sobre els 220Ω per a un corrent de 15mA i 3.3V de tensió d'alimentació.

6.2.3.4 Buzzer [6]

Es tracta d'un transductor electroacústics que converteixen senyals elèctrics en sons. A diferència dels altaveus, aquests són transductors piezoelèctrics que varien el seu volum al ser travessats per corrents elèctrics, fent vibrar així una membrana que és l'encarregada d'emetre el so.



Il·lustració 18. Parts buzzer

Aquest dispositius tenen l'avantatge que són de dimensions reduïdes, amb alta durabilitat i baix consum. Per contra, la qualitat del so es reduïda i tan sols s'utilitzen com a alarmes i dispositius d'avís.

6.2.4 Raspberry Pi 3B

La Raspberry Pi és un ordinador de placa reduïda de baix cost i de potència reduïda en comparació als ordinadors convencionals. Tot i aquestes característiques, s'adapta perfectament a les necessitats del projecte, i és on instal·larem el broker MQTT, OpenHAB, i on rebrem la informació captada pels sensors d'Arduino. A més, enviarà les ordres a Arduino.

El sistema operatiu que utilitzarem s'anomena Raspbian, una distribució de Debian dissenyada per a ser utilitzada en aquests dispositius.



Il·lustració 19. Raspberry Pi 3B

Les característiques més rellevants del dispositiu s'especifiquen a la taula següent:

SOC	BROADCOM (CPU+GPU+DSP+SDRAM+PUERTO USB)	BCM2837
CPU	1.2GHz 64-bit quad-core ARMv8	
SDRAM	1GB (compartits con la GPU)	
Ports USB 2.0	4	
Entrada de vídeo	Connector MPI CSI	
Sortides de vídeo	Connector RCA, HDMI i interfície DSI per a panel LCD	
Sortides d'àudio	Connector de 3.5mm, HDMI	
Emmagatzematge	MicroSD	
Connectivitat de xarxa	10/100 Ethernet via hub USB, Wifi 802.11n, Bluetooth 4.1	

Perifèrics de baix nivell	17 x GPIO y un bus HAT ID
Consum energètic	800mA (4.0W)
Font d'alimentació	5 V vía Micro USB o GPIO header

Taula 3. Característiques Raspberry Pi 3B

6.2.5 OpenVPN

Una xarxa virtual privada (VPN, Virtual Private Network) és una tecnologia de xarxa utilitzada per a connectar-se a una xarxa privada mitjançant una xarxa pública, Internet. Normalment, aquesta tecnologia és utilitzada en l'àmbit empresarial per tal que els seus treballadors puguin accedir als recursos corporatius des de qualsevol lloc, sense necessitat d'estar físicament a l'empresa. A més a més, la VPN permet assegurar confidencialitat i integritat de la informació si s'implementa de manera correcta.

En el projecte, s'utilitza un servei gratuït d'VPN, l'OpenVPN. Aquest servei gratuït de VPN es compon d'un servidor, el qual disposa de connexió a internet dins la xarxa local, i d'un client, el qual pot assumir, mitjançant aquesta xarxa, la mateixa localització geogràfica que el servidor i una direcció IP dins d'aquesta mateixa xarxa local.



Una de les raons de triar aquesta tecnologia, es basa en el fet de la gratuïtat d'aquesta eina, i a banda, la seva facilitat d'ús, la gran seguretat que presenta (xifratge i protecció de confidencialitat) i el suport multiplataforma



Així doncs, per tal de gaudir d'aquesta tecnologia, s'instal·la a la Raspberry un servidor VPN anomenat PiVPN i a l'ordinador amb el qual s'accedirà remotament a la xarxa, un client anomenat OpenVPN-GUI.

Com a component addicional per tal d'accedir a la interfície gràfica del dispositiu Raspberry, instal·larem VNC Viewer a l'ordinador remot, amb la finalitat d'accedir de manera telemàtica al seu escriptori un cop dins de la VPN.

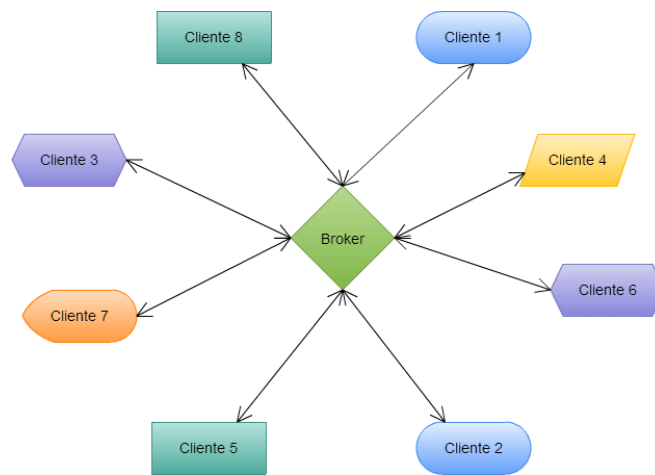


6.2.6 Protocol MQTT

El protocol MQTT és un protocol de comunicació màquina a màquina (M2M) molt utilitzat en l'IoT. Així doncs, serà l'estàndard utilitzat per a comunicar els diferents nodes que conformen el projecte.

Una de les característiques principals que el fan adient per al projecte, és que es tracta d'un protocol que consumeix un ample de banda reduït que funciona a la perfecció en dispositius amb pocs recursos (CPU, RAM...) com els utilitzats.

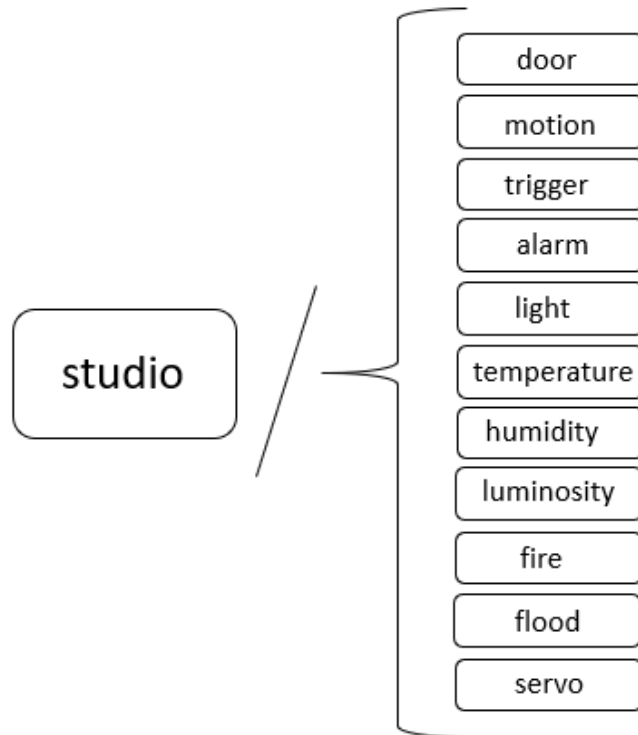
L'arquitectura del protocol segueix una topologia d'estrella amb un node central (Raspberry en el nostre cas) que realitza la funció de servidor o 'broker', i que és capaç de gestionar fins a 10000 clients. La funció del 'broker' és la de gestionar la xarxa i transmetre els missatges. Per la seva banda, els clients envien de manera periòdica un paquet anomenat PINGREQ i esperen la resposta del 'broker' PINGRESP, mantenint així el canal actiu en tot moment.



Il·lustració 20. Funcionament protocol MQTT

La comunicació, d'un a un o d'un a molts, està basada en 'topics', els quals crea un client al publicar un missatge sobre un determinat tema. Per tal de rebre la informació publicada sobre un tòpic, un o més clients receptors s'han de subscriure a ell. Acte seguit, aquest subscriptor pot tornar a publicar sobre aquest mateix 'topic' i convertir-se en emissor, tenint per tant, el caràcter d'emissor i receptor alhora.

Les diferents jerarquies que conformen un 'topic', estan separades pel caràcter /. En el cas del projecte, l'estructura serà la següent:



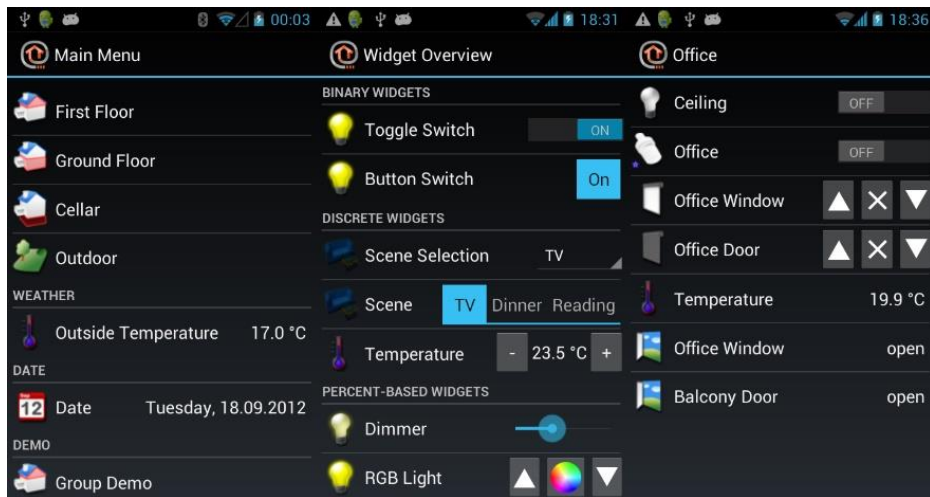
Així per exemple, es podria publicar/subscriure sobre el 'topic' "studio/relay1" en concret, o subscriure's a tots els 'topics' de la manera següent: "studio/#".

6.2.7 OpenHAB

OpenHAB és un software que pretén integrar diferents sistemes de domòtica que permeten automatitzar diferents funcions d'un espai domotitzat (encendré el llum, apagar una televisió...) o controlar de manera telemàtica aquestes funcions, ja sigui des de la xarxa local, o des d'una xarxa externa.

Al tractar-se d'un software desenvolupat en Java, aquest es pot executar sobre qualsevol plataforma que suporti aquesta tecnologia en la seva versió 7 o superior. Així doncs, resulta una tecnologia adient per a ser executada sobre el dispositiu Raspberry.

El software, presenta un entorn visual atractiu capaç de ser personalitzat i adaptar-se a les necessitats de cada projecte en concret, amb la qual cosa resulta efectiu per a projectes amb alt grau d'escalabilitat.



Il·lustració 21. Exemple interfície OpenHAB

Pel que fa al seu funcionament, OpenHAB treballa amb el que anomena ‘items’, que són objectes virtuals que representen instàncies reals com sensors o actuadors, anomenades ‘things’ dins l’entorn del software. Ambdós objectes, es representaran en un mapa virtual del propi entorn de treball, anomenat ‘sitemap’.

Per tal d’evitar la programació via terminal de Raspbian i agilitzar així aquesta tasca, s’utilitza un editor de text enriquit anomenat Notepad++, capaç d’integrar llibreries de llenguatges com els que s’utilitzaran per a OpenHAB (items, sitemaps...).

6.3 Configuració de xarxa

Dins de la xarxa local, s’han definit les següents IP dins el rang estàtic definit al router:

- Raspberry Pi 3B → 192.168.1.10
- Arduino MEGA → 192.168.1.20

Pel que fa a OpenHAB i MQTT, es defineixen els següents ports d’accés sobre la direcció de la Raspberry:

- MQTT: 192.168.1.10:1883
- OpenHAB: 192.168.1.10:8080



Il·lustració 22. Configuració DHCP

7.Implementació

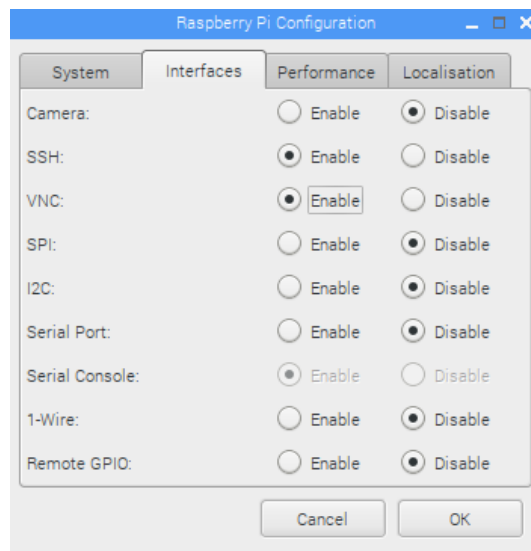
7.1 Configuració Raspberry

S'habilita el protocol SSH a la Raspberry amb la comanda:

```
sudo raspi-config
```

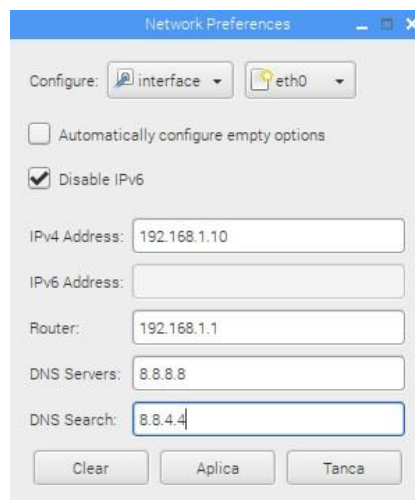
Lavors, dins del menú '*Interfacing options*', seleccionem SSH i l'habilitem.

Tot i que el mètode per comanda funciona, es pot procedir d'una manera més senzilla per permetre la connexió VNC, dirigint-se al menú de configuració de Raspberry, des d'on a més, es pot confirmar que s'ha activat correctament la connexió via SSH:



Il·lustració 23. Activar VNC i SSH a Raspberry

Pel que fa a la configuració de xarxa, s'introdueix un adreçament estàtic amb la direcció del router com a *gateway* i amb els DNS de '*Google Public DNS*':



Il·lustració 24. Configuració xarxa Raspberry

S'observa amb la comanda 'ifconfig' després de fer un 'reboot' com, efectivament, la IP de la Raspberry ha estat canviada:

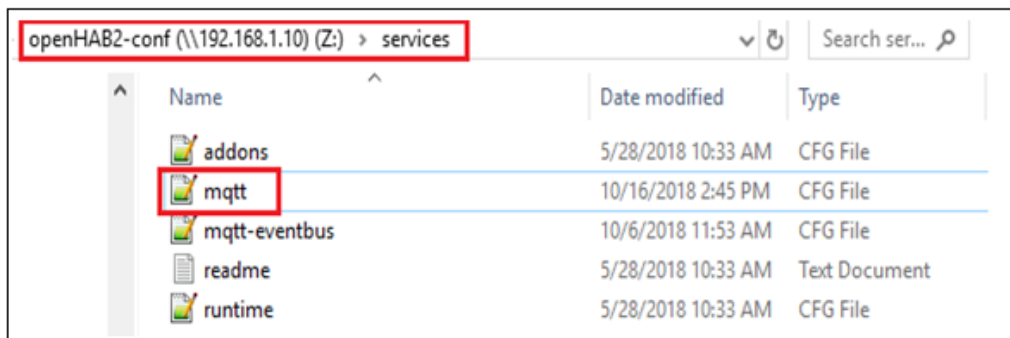
```
pi@raspberrypi:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::ba27:ebff:feff:e537 prefixlen 64 scopeid 0x20<link>
    ether b8:27:eb:f1:e5:37 txqueuelen 1000 (Ethernet)
    RX packets 14 bytes 694 (694.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 28 bytes 4088 (3.9 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Il·lustració 25. Comprovació configuració de xarxa

7.2 Configuració del sistema OpenHAB2

Un cop es disposa de l'entorn OpenHAB instal·lat i funcionant, es procedeix amb la configuració pròpia de l'entorn i el fitxer de configuració de Mosquitto.

En primer lloc, s'edita el fitxer de configuració de Mosquitto i es modifiquen les línies següents, per tal d'indicar la direcció i port del 'broker', l'identificador del servei amb el qual accedir a ell, definir una qualitat de servei determinada i forçar al 'broker' a retenir el missatge enviat:



```
# URL to the MQTT broker, e.g. tcp://localhost:1883 or ssl://localhost:8883
mqtt.url=tcp://192.168.1.10:1883

# Optional. Client id (max 23 chars) to use when connecting to the broker.
# If not provided a random default is generated.
mqtt.clientId=openhab

# Optional. Set the quality of service level for sending messages to this broker.
# Possible values are 0 (Deliver at most once), 1 (Deliver at least once) or 2
# (Deliver exactly once). Defaults to 0.
mqtt.qos=1

# Optional. True or false. Defines if the broker should retain the messages sent to
# it. Defaults to false.
mqtt.retain=true
```

Il·lustració 26. Edició fitxer configuració mosquitto

Tot seguit, s'instal·len el que es coneix com 'bindings', encarregats d'oferir serveis addicionals:

- MQTT Binding: encarregat de la comunicació amb el 'broker' sota el protocol MQTT. OpenHAB actuarà com a client del protocol.
- Network Binding: per detectar dispositius connectats a la xarxa local.
- OpenHAB Cloud: per accedir al servei des de l'exterior de la xarxa local.

Per últim, es presenten les principals característiques dels diferents elements que componen OpenHAB i el seu mode d'edició:

Element	Descripció	Mode d'edició
Things	<p>Representen la capa física (dispositius, serveis web, etc) del sistema OpenHAB. Es connecten a l'aplicació mitjançant 'bindings' (enllaços). Per tant, abans de connectar cap element al sistema, hem de dotar-lo del 'binding' adequat per a que OpenHAB el reconegui i gestioni.</p> <p>A més, cada 'thing' proveeix d'un o més canals per accedir a les seves funcionalitats, les quals es vinculen als 'items' que els representen i permeten controlar-los.</p> <p>A partir d'aquest punt, els 'items' estaran disponibles per a les diferents interfícies d'usuari i per a introduir-los en les regles que definim posteriorment.</p>	Importats a través de PaperUI.
Sitemap	Aquest element és utilitzat per compondre la representació gràfica que l'usuari observarà al connectar-se a qualsevol de les interfícies d'usuari (BasicUI, app, etc).	Notepad++
Items	Representació virtual de la capa física ('things') i les seves funcionalitats. Aquests 'items' es poden agrupar i poden emmagatzemar informació sobre l'estat d'un element del conjunt del sistema.	Notepad++
Rules	En aquest fitxer s'especificaran les diferents automatitzacions del sistema en llenguatge Xtend, basat en Java.	Notepad++

	<p>Així doncs, s'especificaran el seguit de tasques o tasca a realitzar en cas de que succeeixi l'esdeveniment definit. Dins d'aquestes tasques es troben des de realitzar una acció sobre un actuator, fins a enviar una notificació d'alerta al dispositiu mòbil.</p>	
--	---	--

Taula 4. Descripció elements OpenHAB

7.2.1 Definició d'ítems

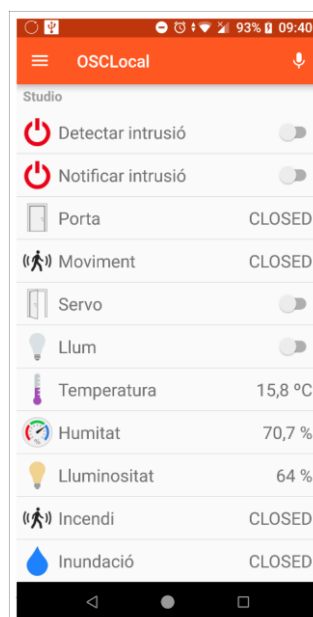
En aquest apartat, es defineix l'estructura per definir els ítems del projecte, que es troben en un fitxer amb extensió '.items' adjuntat en el document.

Així doncs, l'estructura serà la següent:












```
Tipus NomItem "NomEtiqueta [stateformat]" <iconaRepresentativa>
  (grup_pertinençal, grup_pertinença2...) [tag1, tag2...]{configuració del binding}
```

I els ítems definits són: Llum, Porta, Moviment, Temperatura, Humitat, Lluminositat, Sensor de Foc, Alarma, Sensor d'Aigua i Servo.

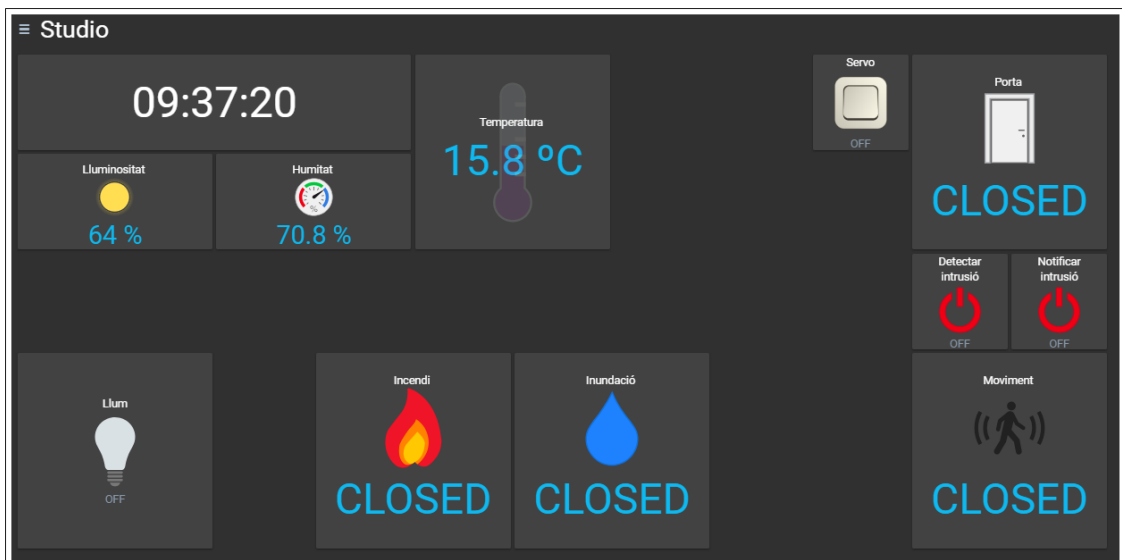
Mitjançant aquests ítems, es pot monitoritzar l'estat de tots i cadascun dels sensors i/o actuadors, per tal de poder actuar, en cas que sigui necessari, sobre ells, o, si és el cas, generar alarmes, les quals es defineixen més tard en aquest document. Aquesta interacció es farà a través de l'aplicació OpenHAB del dispositiu mòbil, de la interfície BasicUI i de la interfície HABPanel, les quals es poden observar en les següents imatges:



Il·lustració 27. App mòbil OpenHAB

Studio					
	Detectar intrusió	<input type="checkbox"/>		Notificar intrusió	<input type="checkbox"/>
	Porta	CLOSED		Moviment	CLOSED
	Servo	<input type="checkbox"/>		Llum	<input type="checkbox"/>
	Temperatura	15,8 °C		Humitat	70,8 %
	Lluminositat	63 %		Incendi	CLOSED
	Inundació	CLOSED			

Il·lustració 28. BasicUI OpenHAB



Il·lustració 29. HABPanel OpenHAB

7.3 Instal·lació i configuració Arduino

7.3.1 Estructura del codi

Dins l'editor de codi de l'IDE d'Arduino es diferencien les següents parts:

Part	Contingut
Inici de codi	<ul style="list-style-type: none"> ▪ S'inclouen les diferents llibreries que s'utilitzen. ▪ Es defineix la configuració de xarxa del dispositiu. ▪ Es defineix la configuració per a treballar sota el protocol MQTT com a client. ▪ Es defineixen les diferents variables, indicant el seu tipus i inicialitzant-les en cas que sigui necessari.
Definició de funcions	<p>Es defineixen el seguit de funcions que utilitzarem, en concret:</p> <ul style="list-style-type: none"> ▪ void reconnect(): funció del protocol MQTT que en cas de fallida

	<p>de connexió, reconnecta al broker o a la xarxa local, depenent quin sigui el problema.</p> <ul style="list-style-type: none"> ▪ void lightRead(): Llegir el sensor de lluminositat. ▪ void dhtRead(): Llegir temperatura i humitat. ▪ void pirRead(): Llegir sensor de moviment en cas de trigger activat de manera continuada. En cas de detecció de moviment, activa el buzzer i el LED i envia un avís al dispositiu mòbil. ▪ void hcsr04Read(): mesurar distància fins a la porta en cas de trigger activat, amb una freqüència de temps de 2 segons. En cas de detecció de porta oberta, activa el buzzer i el LED i envia un avís al dispositiu mòbil. ▪ void mqRead(): Llegir detector d'incendis/gas. En cas de detecció de gas/incendi, activa el buzzer i el LED i envia un avís al dispositiu mòbil. ▪ void waterRead(): Llegir sensor d'inundació. En cas de detecció d'aigua, activa el buzzer i el LED i envia un avís al dispositiu mòbil. ▪ void callback(char* tòpic, byte* payload, unsigned int length): funció del protocol MQTT en la qual, un cop connectat al broker, es reben els missatges als quals s'està subscript i es pren la decisió corresponent en cas que sigui necessari. <p><i>NOTA:</i> El funcionament intern de cadascuna de les funcions, queda definit en l'arxiu amb l'sketch d'Arduino adjuntat a la memòria,</p>
Funció setup	<ul style="list-style-type: none"> ▪ S'inicien les diferents variables i la configuració inicial. ▪ Aquesta part del codi s'executa un sol cop.
Funció loop	<ul style="list-style-type: none"> ▪ Part del codi que s'executa contínuament. ▪ Es llancen les funcions definides amb anterioritat.

Taula 5. Descripció de blocs sketch Arduino

Així doncs, es treballa en mode superloop, en el qual les diferents accions es van repetint de manera indefinida. Amb aquesta metodologia, es facilita la programació però a canvi es perd control sobre els possibles retards i temps de resposta. Per pal·liar les característiques negatives obtingudes amb el mètode de repetició infinita, s'utilitza la funció millis() [1], amb la qual es pot controlar els temps de lectura i publicació dels diferents sensors, definint intervals de temps per tal d'assolir els principals objectius del projecte. Evitant així retards crítics i la saturació del processador.

A més a més, es fa ús de les interrupcions pròpies d'Arduino [19] pel cas dels sensors de inundació i incendi. Amb aquestes interrupcions, les funcions waterRead i mqRead tan sols s'executaran en cas de que es detecti un canvi al pin d'entrada on estan connectats els sensors, la qual cosa suposa un estalvi d'energia i de treball per part d'Arduino. En concret, s'utilitzen els pins 20 i 21, els quals disposen d'aquesta funcionalitat. Un cop finalitzada l'execució de la interrupció, es torna al flux normal del programa, el qual s'atura en executar-se aquest tipus d'accions.

7.3.2 Arduino i Ethernet Shield WS5100

Amb aquest element, dotarem de connexió a la xarxa al dispositiu Arduino.

Per tal que això sigui possible, cal incloure dins el programa la llibreria 'Ethernet.h'. A més, cal dotar Arduino d'una direcció MAC i la configuració de xarxa adjuntada en la fotografia:

```
// Configuració Ethernet
byte mac[] = { 0xDE, 0xED, 0xBF, 0xAF, 0xFF, 0xED };
IPAddress ip(192, 168, 1, 20);
IPAddress subnet(255, 255, 255, 0);
IPAddress gateway(192, 168, 1, 1);
```

Dins de cada iteració de la funció 'loop', i tan sols en cas que el dispositiu no es trobi connectat a la xarxa prèviament, es realitzarà la connexió.

7.3.3 Arduino i MQTT

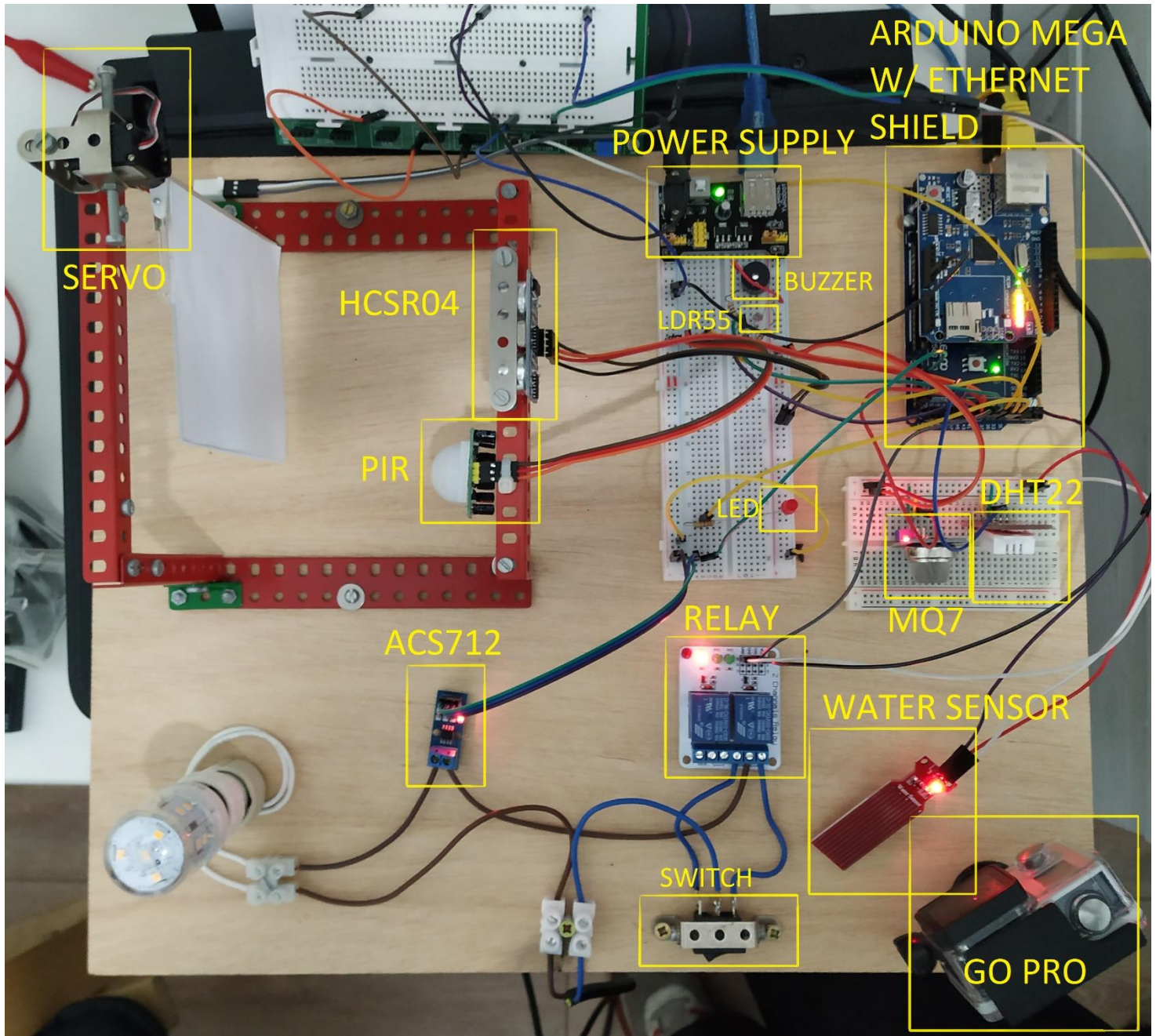
Per tal de permetre la comunicació sota el protocol MQTT i que el dispositiu Arduino sigui capaç de publicar i subscriure's als diferents tòpics com a client, cal incloure la llibreria 'PubSubClient.h'.

A més a més, cal definir els següents elements:

- IP del 'broker' MQTT i establiment de client de la llibreria PubSubClient.h:
 - ✓ IPAddress mqtt_server(192,168,1,10)
- Variable que determini una instància client dins del protocol de comunicació:
 - ✓ PubSubClient client(ethClient);
- Funcions del protocol MQTT (reconnect i callback) definides anteriorment.
- Iniciar variables per a establir-se com a client dins la funció setup, en concret:
 - ✓ client.setServer(mqtt_server, 1883);
 - ✓ client.setCallback(callback);

7.4 Maqueta i interconnexió de sensors i actuadors

Per tal de comprovar el funcionament del sistema i poder plasmar el projecte físicament, es crea un entorn de proves en forma de maqueta. En la següent fotografia, s'observen els diferents elements que la componen:

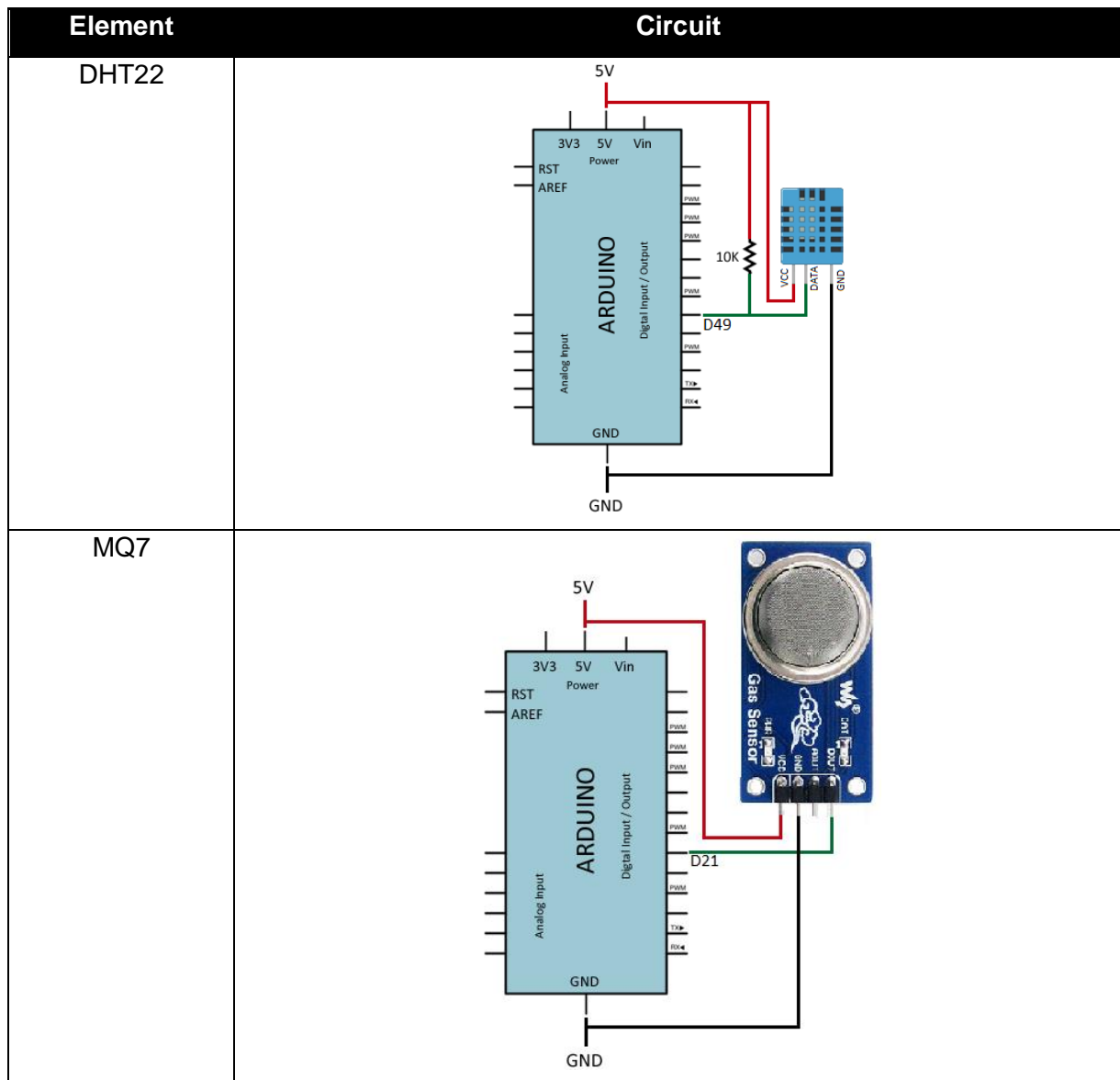


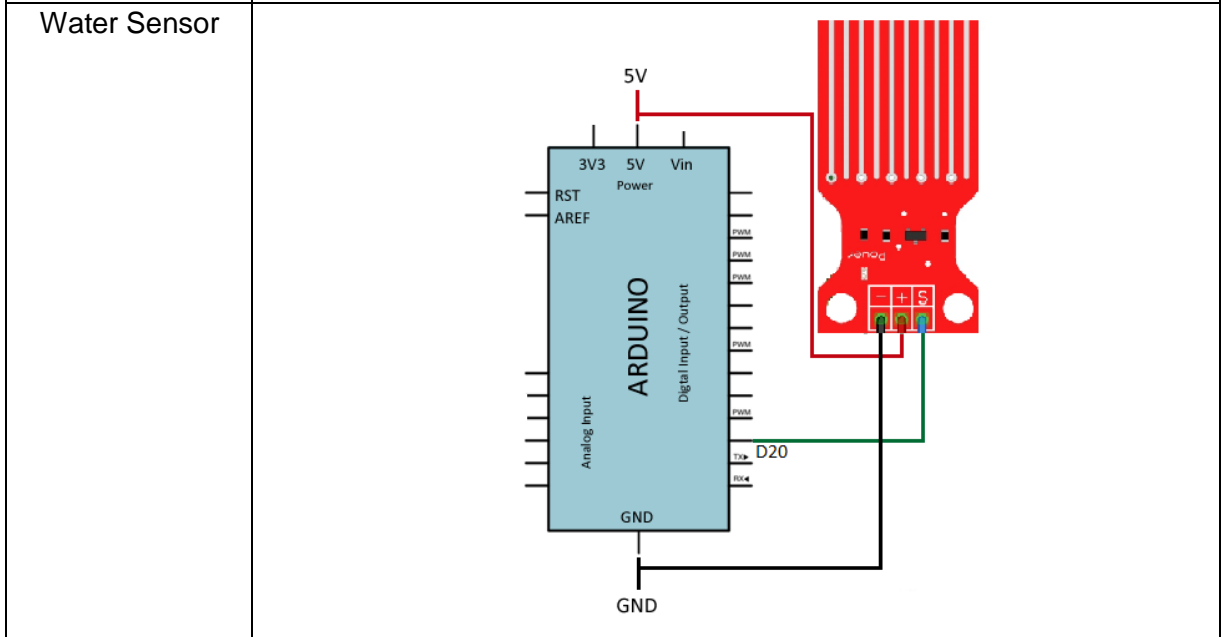
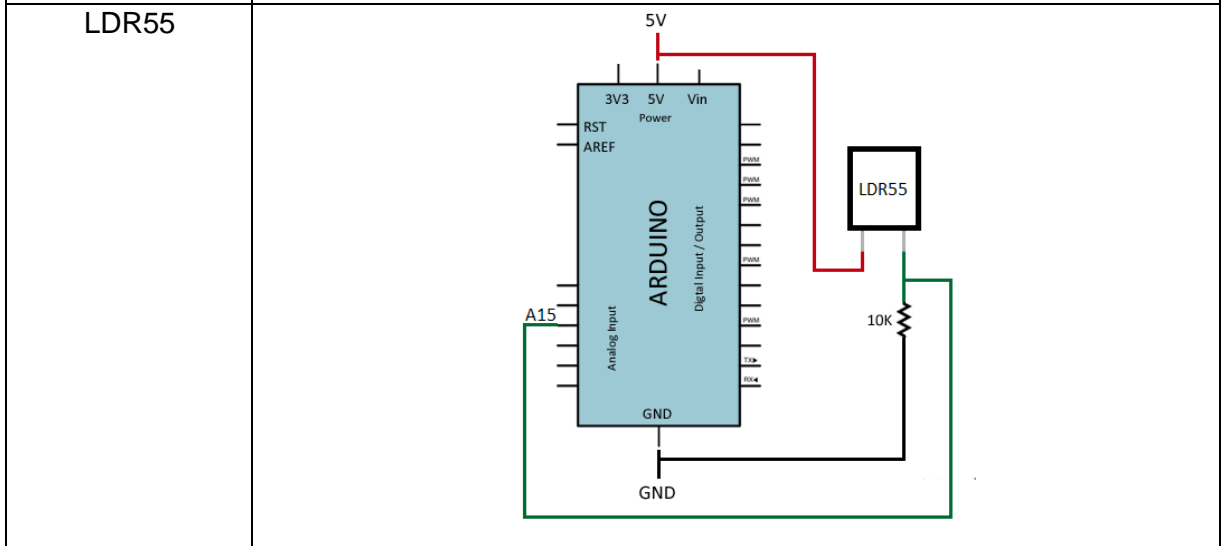
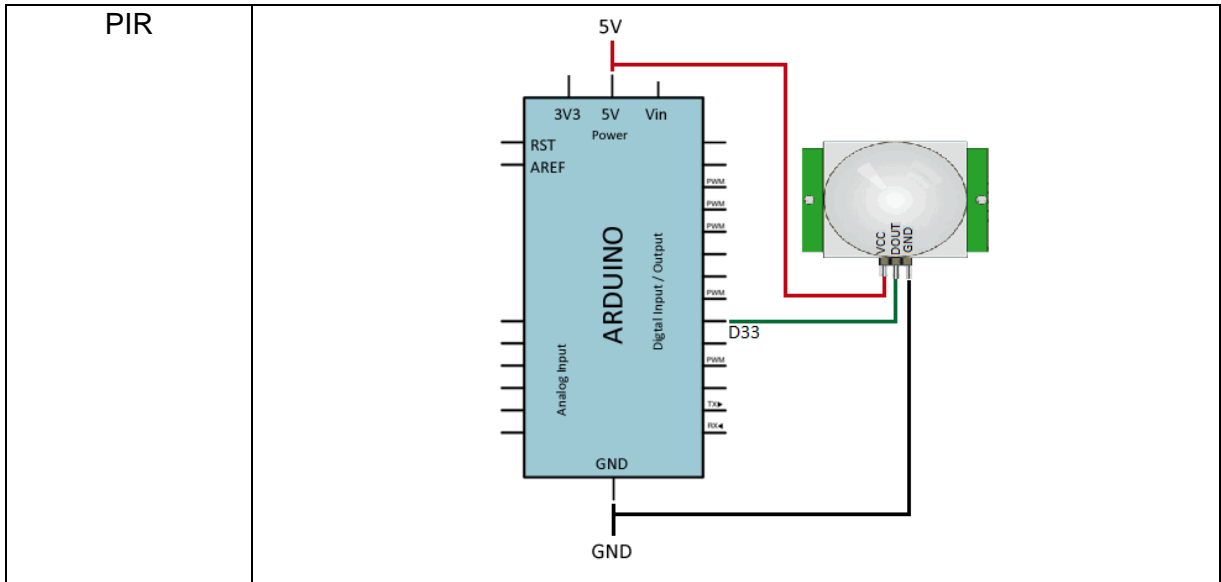
Il·lustració 30. Maqueta projecte

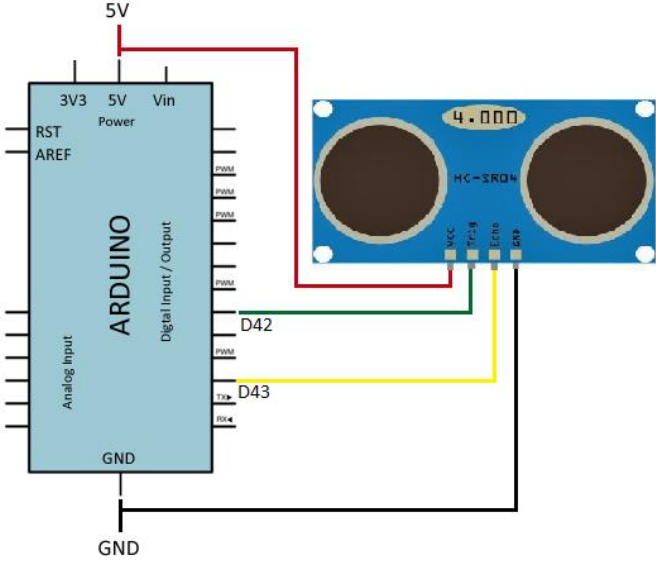
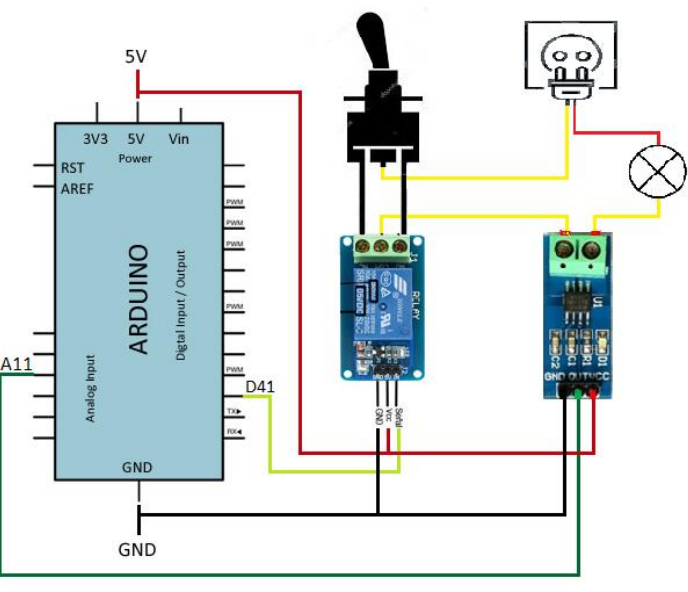
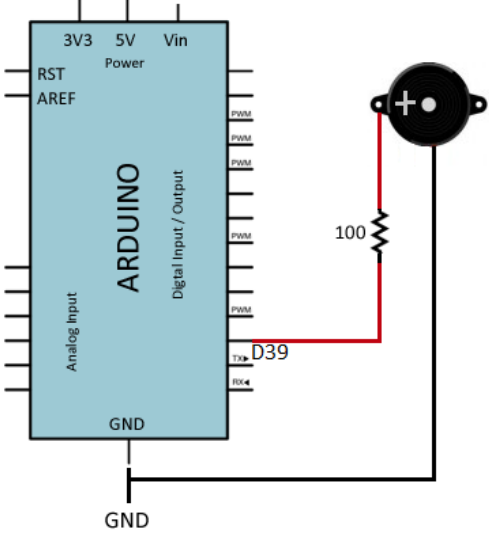
7.4.1 Interconnexió de sensors i actuadors

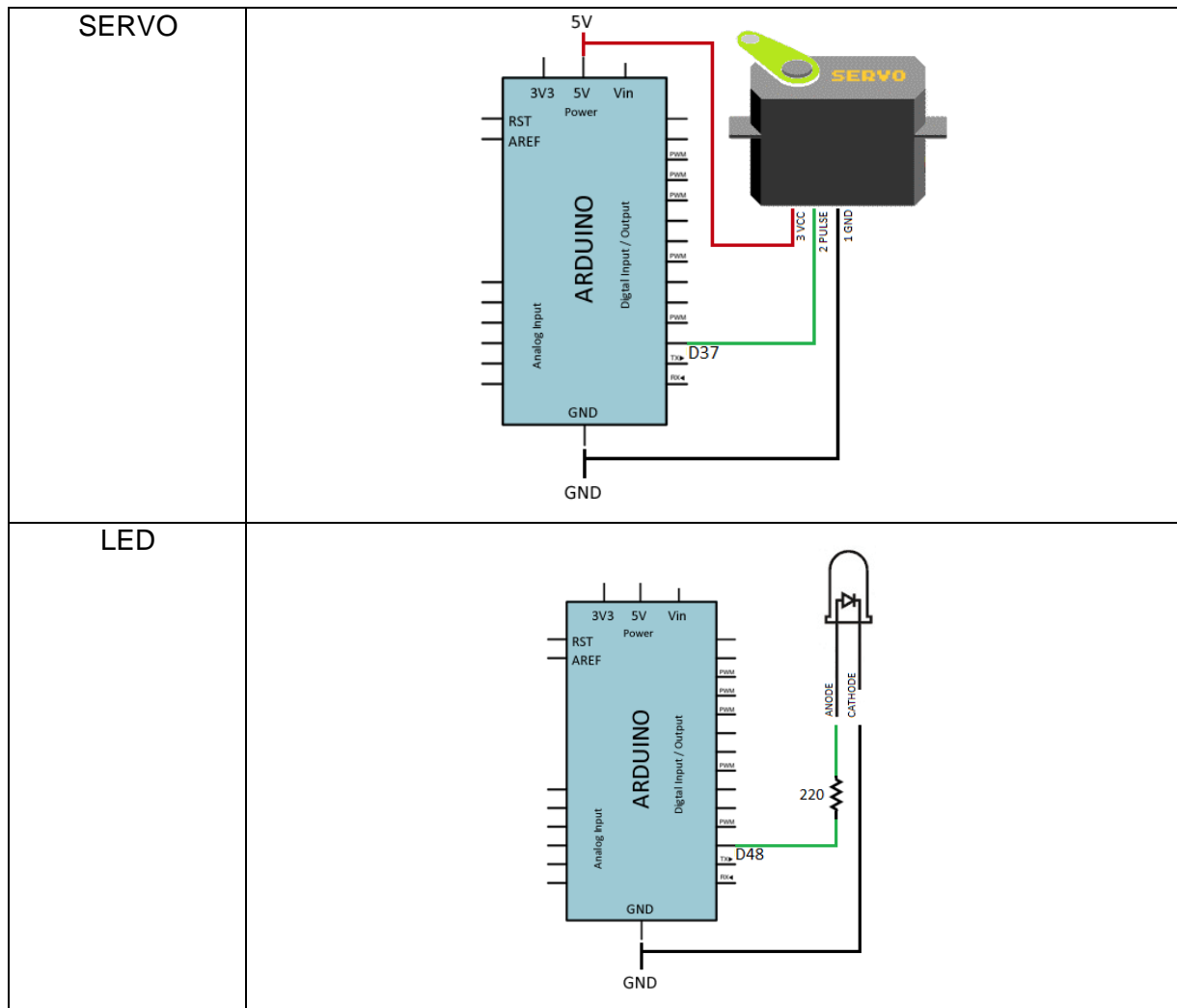
Pel que fa a l'alimentació de corrent elèctric a Arduino i a la protoboard, i per consegüent, tots els elements del sistema,, ja s'ha mencionat amb anterioritat que es fa ús del dispositiu Power Supply, per tant, en aquest apartat es definiran de manera esquemàtica i funcional

les connexions establides en la maqueta per tal de facilitar la comprensió a nivell circuit del projecte:





<p>HC-SR04</p>	
<p>ACS712 + RELÉ + COMMUTADOR</p>	
<p>BUZZER</p>	



Taula 6. Esquema circuital dels diferents elements

Juntament amb aquesta configuració, es defineixen les llibreries que s'utilitzaran, les constants i els ports d'entrada/sortida de la següent manera:

```

#include <DHT.h>
#include <Servo.h>

#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321
const int LDRPin = A15;
const int DHTPin = 49;
const int PIRPin= 33;
const int ECHOPin = 43;
const int TRIGGERPin = 42;
const int BUZZPin = 39;
const int MQPin = 21;
const int LEDPin = 31;
const int WATERPin = 20;
const int RELAYPin = 41;
const int ACS712Pin = A11;
const int SERVOPin = 37;
DHT dht(DHTPin, DHTTYPE);
Servo servoPorta;

void setup() {
  Serial.begin(9600);
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
  servoPorta.attach(37);
  pinMode(LDRPin, INPUT);
  pinMode(DHTPin, INPUT);
  pinMode(PIRPin, INPUT);
  pinMode(ACS712Pin, INPUT);
  pinMode(TRIGGERPin, OUTPUT);
  pinMode(LEDPin, OUTPUT);|
  pinMode(ECHOPin, INPUT);
  pinMode(BUZZPin, OUTPUT);
  pinMode(RELAYPin, OUTPUT);
  pinMode(MQPin, INPUT_PULLUP);
  pinMode(WATERPin, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(WATERPin), waterRead, CHANGE);
  attachInterrupt(digitalPinToInterrupt(MQPin), mqRead, CHANGE);
}

```

7.5 Definició regles OpenHAB

Per tal de rebre avisos sobre esdeveniment que es considerin importants i necessaris de coneixes amb prioritat alta al nostre dispositiu mòbil, OpenHAB disposa de documents amb extensió '.rules' que permeten configurar regles d'alarmes segons els nostres interessos.

La sintaxi d'aquest tipus de fitxers consta de la següent estructura:

```

rule "NOM DE LA REGLA"
when
  //CONDICIÓ DE TRIGGER: en el nostre cas, en cas que l'item
  //esmentat rebí una actualització
then
  //ACCIONS A DUR A TERME: en el nostre cas, enviarà una notificació al
  //dispositiu mòbil en cas de complir-se la condició
end

```

En el projecte, es defineixen les alarmes següents:

- Alarma de detecció d'intrús, la qual s'activarà si el topic 'Alarm' es troba en estat ON i es detecta moviment:

```
rule "Moviment detectat"
when
    Item Motion received update
then
    if (Motion.state == OPEN && Alarm.state == ON){
        sendNotification("oscar.solans@gmail.com", "MOVIMENT DETECTAT!!!");
    }
end
```

- Alarma de detecció de porta oberta, la qual s'activarà si el topic 'Alarm' es troba en estat ON i es detecta la porta oberta:

```
rule "Porta oberta"
when
    Item Door received update
then
    if (Door.state == OPEN && Alarm.state == ON){
        sendNotification("oscar.solans@gmail.com", "PORTA OBERTA!!!");
    }
end
```

- Alarma de detecció d'inundació, la qual s'activarà en cas que el sensor 'Water Sensor' detecti al presència d'aigua:

```
rule "Inundació"
when
    Item Flood received update
then
    if (Flood.state == OPEN){
        sendNotification("oscar.solans@gmail.com", "INUNDACIÓ!!!");
    }
end
```

- Alarma de detecció d'incendi, la qual s'activarà en cas que el sensor MQ-7 detecti foc o gasos perillosos:

```
rule "Detecció d'incendi"
when
    Item Fire received update
then
    if (Fire.state == OPEN){
        sendNotification("oscar.solans@gmail.com", "INCENDI DETECTAT!!!");
    }
end
```

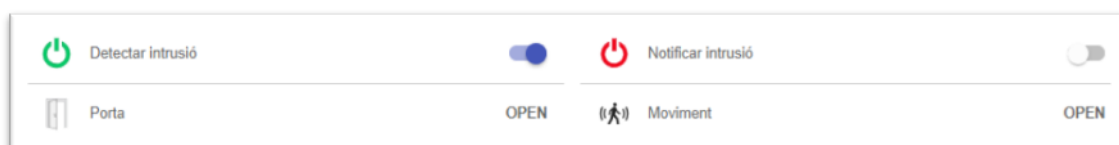
8. Comprovació i resultats

8.1 Comprovació del sistema

Per tal de comprovar el correcte funcionament del projecte un cop aquest ja està completament implementant i en funcionament, es realitzen un seguit de proves amb la finalitat de corroborar que els resultats esperats han estat assolits.

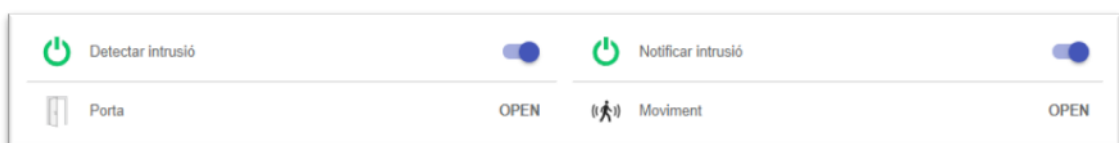
Les proves a realitzar seran:

1. Activant el switch "Detectar intrusió", la qual cosa equival a situar el trigger a 1, els sensors de detecció de moviment i detector de porta oberta comencen a prendre mesures:



Hi ha moviment
La distància és de 81cm

2. Si al cas anterior, li afegim l'activació del switch "Notificar intrusió", en cas de que qualsevol d'aquests dos sensors es situïn en valor "OPEN", la qual cosa equival a detectar una intrusió o porta oberta (distància >11cm), s'envia una notificació al dispositiu mòbil, alertant quin dels dos sensors ha detectat quelcom. En la prova, s'obrirà la porta per tal d'afectar ambdós sensors:

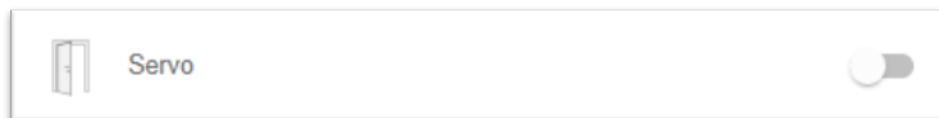
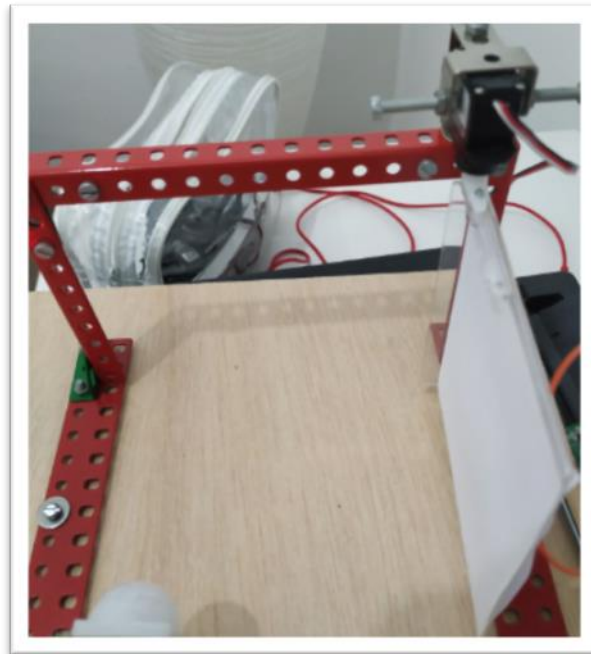


openHAB • ara ▾
openHAB MOVIMENT DETECTAT!!!
openHAB PORTA OBERTA!!!
Gestiona les notificacions ESBORRA-HO TOT

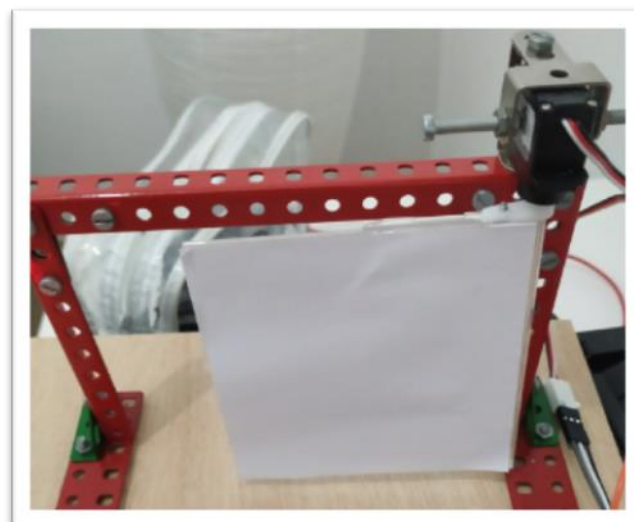
3. Activant i desactivant el switch "Servo", s'obre i es tanca la porta respectivament:



Servo: 1



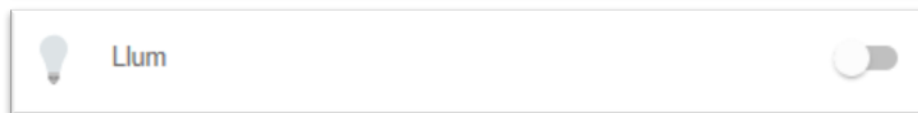
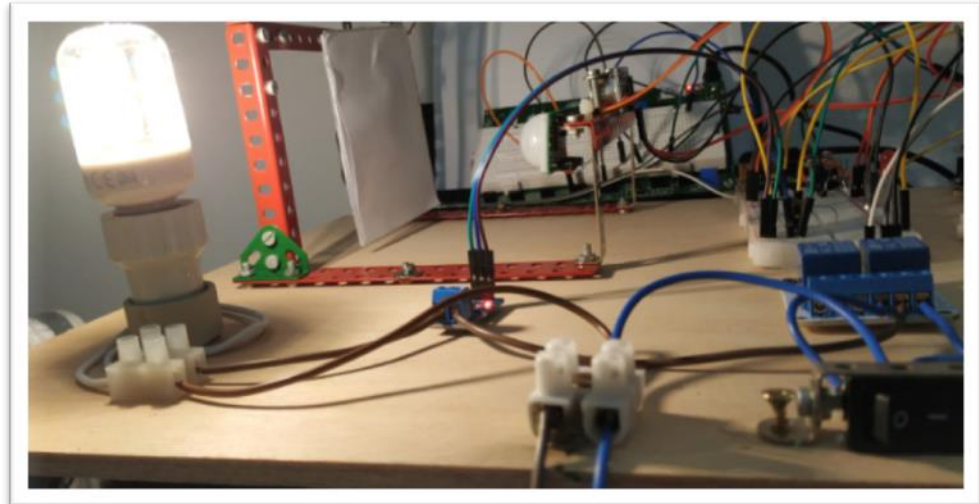
Servo: 0



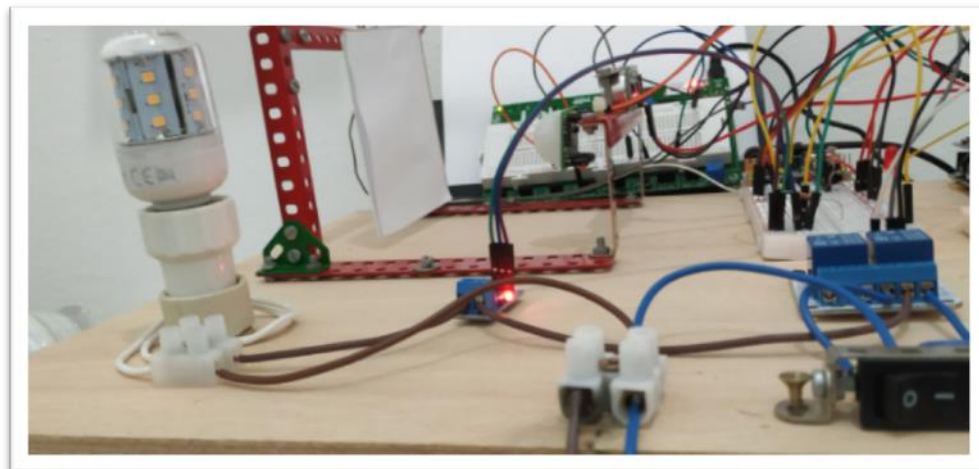
4. Activant i desactivant el switch "Llum", s'obre i es tanca respectivament el que tenim endollat al relé (una bombeta en el cas del projecte) mantenint en la mateixa posició el commutador:



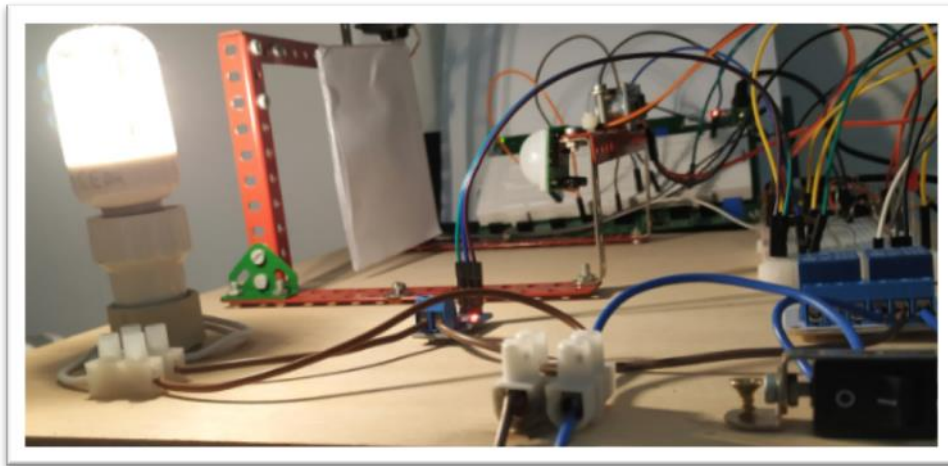
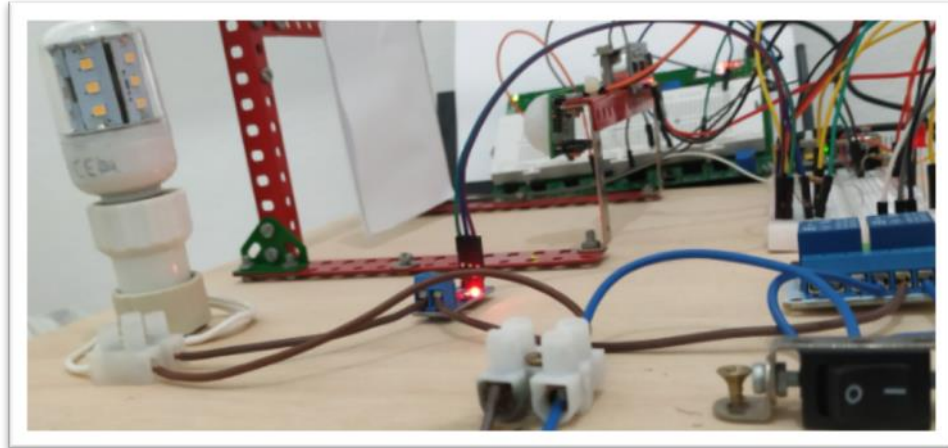
Relé: 1



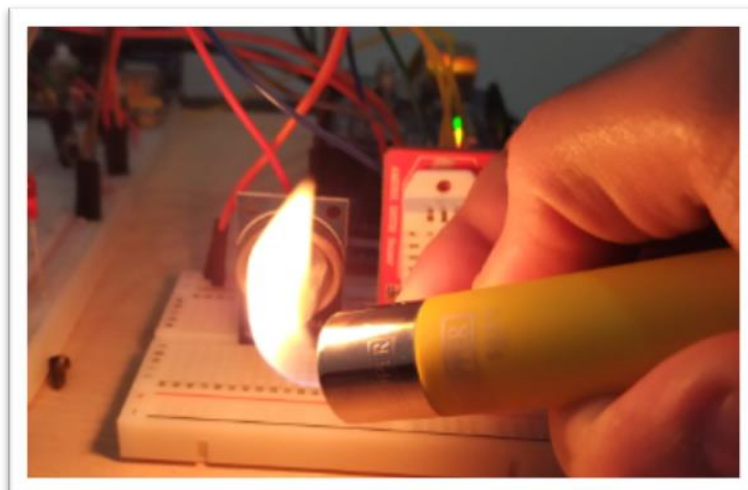
Relé: 0



5. En cas d'activació o desactivació de la bombeta del cas anterior mitjançant el commutador físic, aquest switch canvia automàticament de valor.



6. Apropant una flama o gas al detector d'incendi, aquest es situa en valor "OPEN" i envia una notificació el dispositiu mòbil alertant d'incendi.



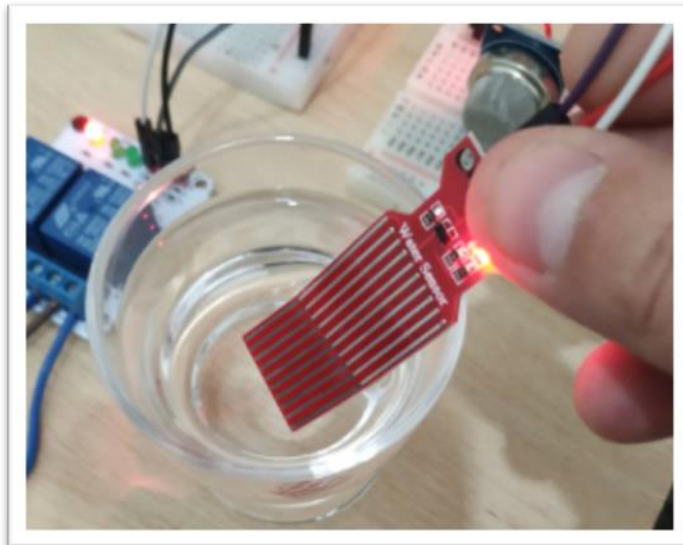
 Incendi OPEN

Hi ha foc!!!

 openHAB • ara
openHAB
INCENDI DETECTAT!!!


Gestiona les notificacions **ESBORRA-HO TOT**

7. Submergint en aigua el sensor d'inundació, aquest es situa en valor "OPEN" i envia una notificació el dispositiu mòbil alertant d'inundació:



 Inundació OPEN

Inundacio!!!

 openHAB • ara
openHAB
INUNDACIÓ!!!

Gestiona les notificacions **ESBORRA-HO TOT**

8.2 Resultats i conclusions

Retornant als objectius mencionats a l'inici del document, es valoren els resultats obtinguts en la següent taula:

Implementar un sistema domòtic que es comuniqui sota el protocol MQTT amb missatges que requereixen poc ample de banda.	✓
Utilitzar l'aplicació opensource OpenHAB per interactuar amb el sistema implementat.	✓
Gestionar una xarxa VPN per accedir al sistema amb independència del sistema natiu d'OpenHAB.	✓
Utilitzar interrupcions d'Arduino en cas de detecció d'inundació o foc, per lliurar l'Arduino de temps de computació i estalviar energia..	✓
Configurar les alarmes i la seva conseqüent notificació al dispositiu mòbil.	✓

Taula 7. Resultats del projecte

Efectivament, es comprova que els objectius marcats per al projecte s'han assolit complint amb el pressupost i calendari marcats. L'ús de tecnologies opensource, a més, ha marcat la diferència en l'elaboració del mateix.

Pensant en possibles millores que s'han vist reflectides durant el desenvolupament del mateix, es llisten les següents:

- Distribució de càrrega de treball en diferents nodes Arduino, cadascun encarregat d'un nombre menor de sensors o actuadors, i per tant, de menys computació.
- Nodes Arduino amb antena WiFi per tal d'evitar cablejat ethernet.
- En paral·lel, disposar d'un sistema redundat per evitar pèrdua de control en cas de fallida del sistema principal i reduir el temps de recuperació del sistema.

En conclusió, podem dir que el ventall de possibilitats observades durant el projecte, fa que el sistema disposi d'una gran escalabilitat i capacitat d'adaptació als requisits de l'usuari final, a un cost reduït i amb termini d'implantació de curta durada.

Bibliografía

- [1] Norwegian Creations - Millis; <https://www.norwegiancreations.com/2017/09/arduino-tutorial-using-millis-instead-of-delay/>
- [2] Luis Llamas - Fotorresistencia LDR ; <https://www.luisllamas.es/medir-nivel-luz-con-arduino-y-fotorresistencia-ldr/>
- [3] Luis Llamas - DHT22; <https://www.luisllamas.es/arduino-dht11-dht22/>
- [4] Luis Llamas – Sensor PIR; <https://www.luisllamas.es/detector-de-movimiento-con-arduino-y-sensor-pir/>
- [5] Luis Llamas – Sensor HC-SR04; <https://www.luisllamas.es/medir-distancia-con-arduino-y-sensor-de-ultrasonidos-hc-sr04/>
- [6] Instructables – Buzzer; <https://www.instructables.com/id/How-to-use-a-Buzzer-Arduino-Tutorial/>
- [7] Naylamp Mechatronics – MQ7; https://naylampmechatronics.com/blog/42_Tutorial-sensores-de-gas-MQ2-MQ3-MQ7-y-MQ13.html
- [8] Luis Llamas – WaterSensor; <https://www.luisllamas.es/arduino-humedad-suelo-fc-28/>
<https://www.prometec.net/sensor-agua/>
- [9] Luis Llamas - ACS712; <https://www.luisllamas.es/arduino-intensidad-consumo-electrico-ac712/>
- [10] Naylamp Mechatronics – ACS712; https://naylampmechatronics.com/blog/48_tutorial-sensor-de-corriente-ac712.html
- [11] Luis Llamas – Relé; <https://www.luisllamas.es/arduino-salida-rele/>
- [12] Prometec – Relé; <https://www.prometec.net/relés/>
- [13] Domoprac – Domótica; <http://www.domoprac.com/protocolos-de-comunicacion-y-sistemas-domoticos/historia-de-la-domotica-pasado-presente-y-futuro/1.-introduccion-la-revolucion-domotica.html>
- [14] Domotizados – Domótica; <https://domotizados.co/conoce-los-inicios-de-la-domotica/>
- [15] Arqcompus – Domótica; <http://arqcompus-domotica.blogspot.com/2009/06/arquitectura.html>
- [16] Geeky Theory – MQTT; <https://geekytheory.com/que-es-mqtt>

[17] Java Hispano – OpenHAB; <http://www.javahispano.org/portada/2014/10/20/openhab-el-software-java-para-domotica.html>

[18] RelarDroid – NoIP; <https://www.realdroid.es/2016/10/29/configurar-no-ip-para-raspberry-pi-y-de-paso-que-es-no-ip/>

[19] Luis Llamas – Interrupciones Arduino; <https://www.luisllamas.es/que-son-y-como-usar-interrupciones-en-arduino/>

[20] Notepad++ – OpenHAB; <https://www.openhab.org/docs/configuration/editors.html>

[21] VNC Viewer Tutorial; <https://archive.realvnc.com/products/vnc/documentation/4.2/win/winvncviewer.pdf>

[22] Raspberry Pi – OpenHAB; <https://www.openhab.org/docs/installation/rasppi.html>

[23] OpenGUI (Windows); <https://community.openvpn.net/openvpn/wiki/OpenVPN-GUI>

ANNEX I - Instal·lació Raspberry

Què es necessita?

Amb la compra de la Raspberry Pi 3, tan sols s'inclou la pròpia controladora, així doncs, addicionalment es necessita:

- Alimentador de 5V i 2.5A amb sortida microUSB.
- Targeta de memòria microSD.
- Ratolí i teclat USB.
- Pantalla amb HDMI.
- Cable HDMI.
- Instal·lador Noobs per al sistema operatiu Raspbian.

Instal·lació sistema operatiu Raspbian

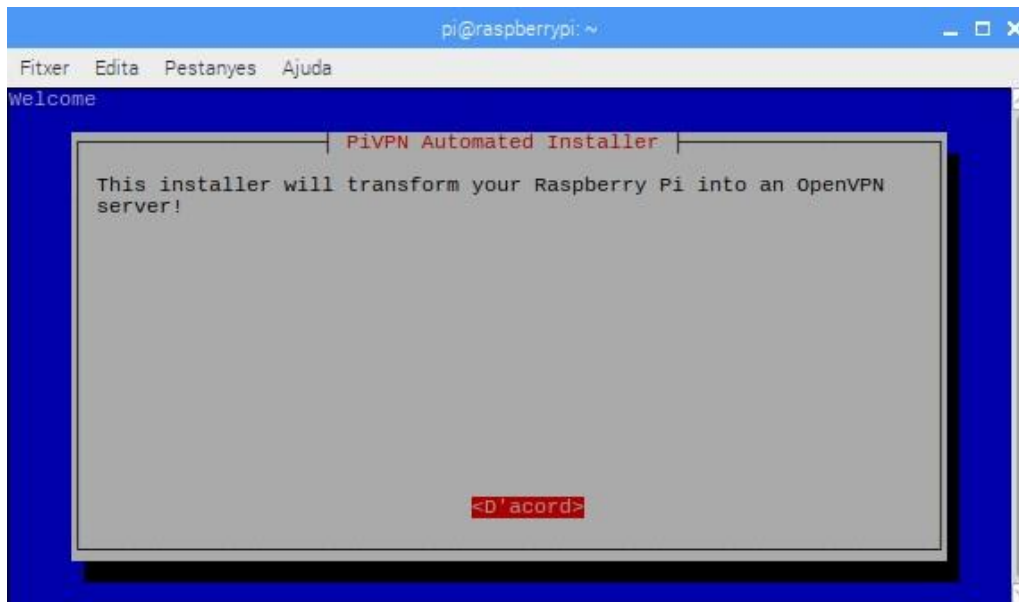
Un cop es tenen tots els elements necessaris, es pot procedir realitzant un format de la targeta microSD amb el programa '*SD Card Formater*' per exemple, per tal d'esborrar qualsevol possible contingut que pugui emmagatzemar.

Així doncs, es procedeix amb la instal·lació del sistema operatiu Raspbian mitjançant Noobs, el qual proporciona un sistema de fitxers expandit, adequat per a la instal·lació d'openHAB.

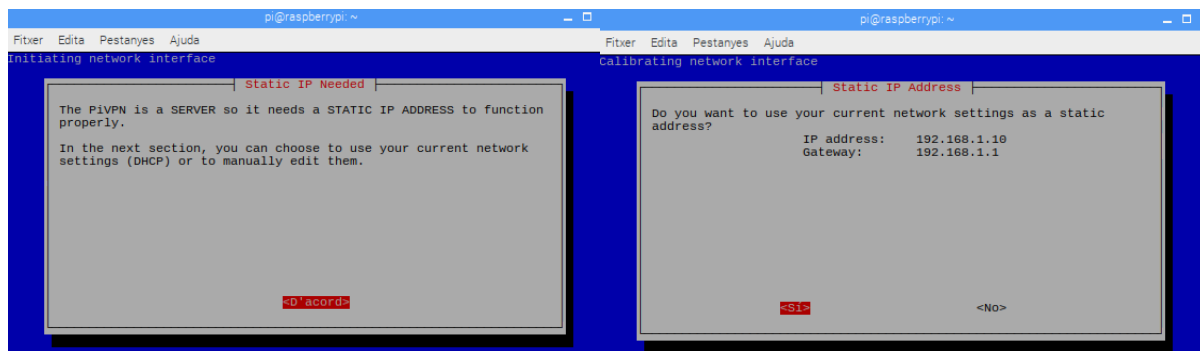
ANNEX II - Instal·lació i configuració PiVPN

Per instal·lar PiVPN per tal de poder accedir a la xarxa local des de qualsevol xarxa via VPN (*Virtual Private Network*), s'introdueix la següent comanda a un terminal, la qual executarà un *script* del web directament al *bash* de la nostra Raspberry:

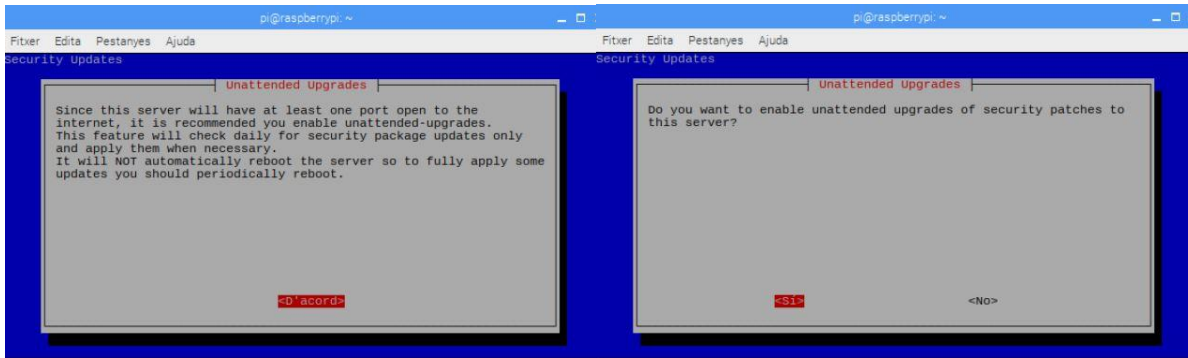
```
curl -L https://install.pivpn.io | bash
```



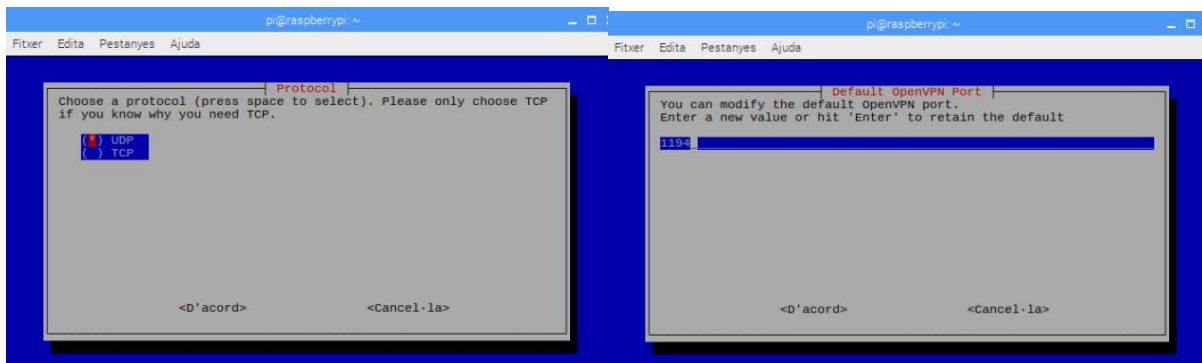
Per tal que el sistema funcioni correctament cal un adreçament estàtic que el propi instal·lador permet configurar, tot i que, com ja s'ha configurat manualment, es pot mantenir la configuració personalitzada prèviament:



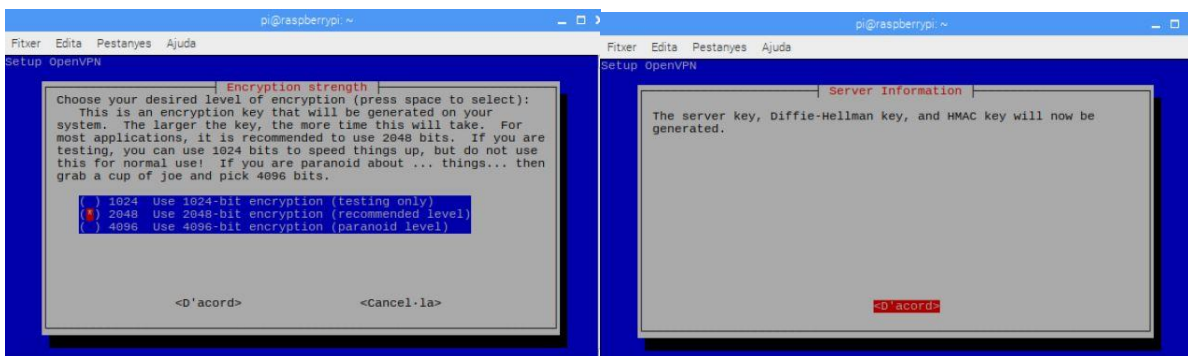
L'instal·lador avisa que les actualitzacions de seguretat seran instal·lades tan sols quan aquestes siguin necessàries en cas que s'activi l'escolta d'actualitzacions pendents, per tant, s'activa aquesta opció:



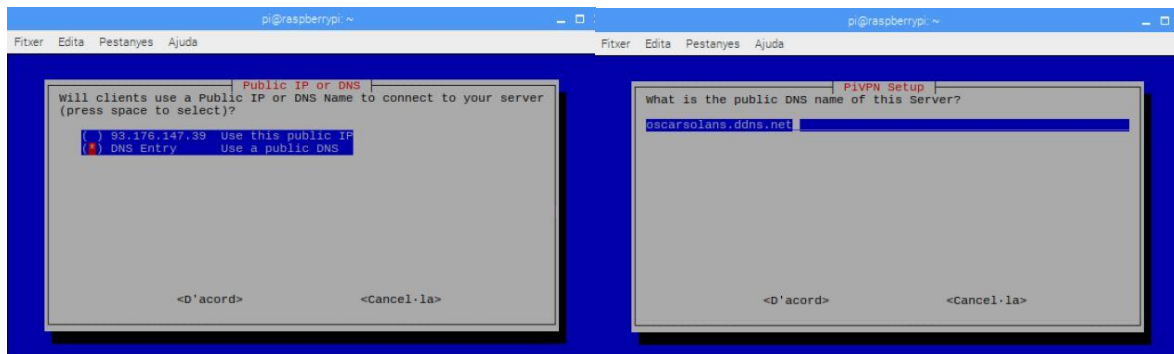
Acte següent, s'introdueix el protocol de comunicació UDP i el port 1994 com a port per aquest servei:



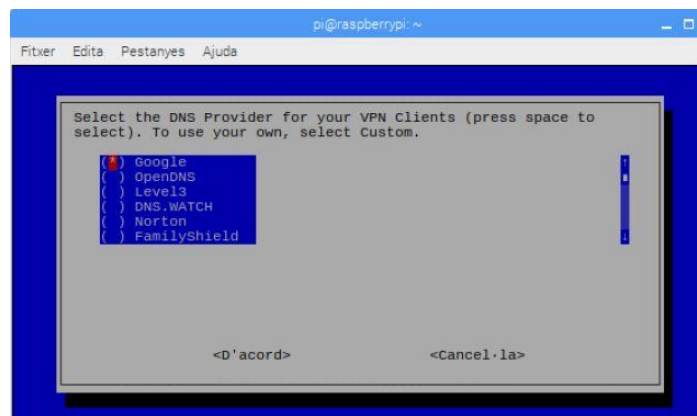
La següent pantalla, indica quants bits es volen utilitzar per l'encryptació de la clau i es marca 2048 seguint les recomanacions de l'instal·lador, ja que es considera un nivell de seguretat acceptable. A més, també informa de l'algoritme utilitzar per a generar la clau:



El següent pas pren una forta rellevància, i és que cal indicar si els usuaris es connectaran a través d'una IP concreta (per defecte la IP pública que tenim en aquell moment) o bé si es connectaran a través d'un domini DNS, opció triada ja que posteriorment, es crea un domini des de la pàgina de NoIP (procediment explicat posteriorment en aquest document):



En el següent pas, es tria el servidor DNS de Google i s'afegeix l'usuari 'uoc', el qual podrà accedir a la xarxa OpenVPN i configurar-la. Per fer-ho, s'introdueix la comanda *pivpn add* a un terminal de Raspbian:



```

root@raspberrypi:/home/pi# pivpn add
Enter a Name for the Client: uoc
Enter the password for the client:
Enter the password again to verify:
spawn ./easyrsa build-client-full uoc

Note: using Easy-RSA configuration from: ./vars
rand: Use -help for summary.
Generating a 2048 bit RSA private key
++++
.....++++
writing new private key to '/etc/openvpn/easy-rsa/pki/private/uoc.key.7XlTCjUpw'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
Using configuration from /etc/openvpn/easy-rsa/openssl-1.0.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName            :ASN.1 12:'uoc'
Certificate is to be certified until Sep 29 20:01:52 2028 GMT (3650 days)

Write out database with 1 new entries
Data Base Updated
spawn openssl rsa -in pki/private/uoc.key -aes128 -out pki/private/uoc.key
Enter pass phrase for pki/private/uoc.key:
writing RSA key
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
Client's cert found: uoc.crt
Client's Private Key found: uoc.key
CA public Key found: ca.crt
tls-auth Private Key found: ta.key

=====
Done! uoc.ovpn successfully created!
uoc.ovpn was copied to:
/home/pi/ovpns
for easy transfer. Please use this profile only on one
device and create additional profiles for other devices.
=====

```

Per últim, es configura el router per permetre el tràfic a la Raspberry pel port configurat, 1194:

dirección IP estática									
aplicación / servicio	protocolo	dirección IP origen	dirección de la máscara IP	puerto de origen	dirección IP destino	dirección de la máscara IP	puerto de destino	acción	
openHAB ▼	UDP ▼	<input type="text"/>	<input type="text"/>	1194	192.168.1.10	255.255.255.0	1194	acepta ▼	<input type="button" value="guardar"/>

ANNEX III - Instal·lació OpenHAB

Per tal d'instal·lar openHAB 2 [17] [22] a la Raspberry, primer cal que es configuren certs paràmetres de la Raspberry (<https://www.openhab.org/docs/installation/rasppi.html>) amb la comanda '*sudo raspi-config*' (per realitzar les comandes, podem utilitzar SSH o VNC i un terminal):

- Es canvia el *hostname* a 'raspberry'.
- Es defineix a 16MB la memòria GPU, valor suficient tenint en compte la utilització insignificant de memòria gràfica que realitza el dispositiu Raspberry.

Es reinicia i s'actualitzen tots els paquets existents a la màquina amb les comandes:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

A continuació, s'instal·la un entorn Java, per exemple, l'entorn Zulu Embedded (<https://community.openhab.org/t/howto-install-zulu-embedded-java-on-raspberry-pi-3/22589>), ja que és condició necessària per a executar openHAB 2:

```
sudo su
```

```
apt-get install dirmngr
```

```
apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv-keys 0x219BD9C9
```

```
echo 'deb http://repos.azulsystems.com/debian stable main' >  
/etc/apt/sources.list.d/zulu.list
```

```
apt-get update -qq
```

```
apt-get install zulu-embedded-8
```

I es comprova que s'ha realitzat correctament la instal·lació, que es disposa de la versió actualitzada i que és la que s'utilitza per defecte:

```
java -version
```

```
openjdk version "1.8.0_152"  
OpenJDK Runtime Environment (Zulu Embedded 8.25.0.76-linux-aarch32hf) (build 1.8  
.0_152-b76)  
OpenJDK Client VM (Zulu Embedded 8.25.0.76-linux-aarch32hf) (build 25.152-b76, m  
ixed mode, Evaluation)
```

```
update-alternatives --config java
```

```

Hi ha 2 possibilitats per a l'alternativa java (que proveeix /usr/bin/java).

Selecció   Camí                                     Prioritat  E
stat
-----
* 0        /usr/lib/jvm/zulu-embedded-8-armhf/bin/java 1082500
mode automàtic
  1        /usr/lib/jvm/jdk-8-oracle-arm32-vfp-hflt/jre/bin/java 318
mode manual
  2        /usr/lib/jvm/zulu-embedded-8-armhf/bin/java 1082500
mode manual

Premeu retorn per a mantenir l'opció per defecte[*], o introduïu un número de se
lecció: █

```

Tot seguit, es permet l'ús del protocol HTTPS per a 'apt' i s'instal·len el repositori estable de Bintray, dos requisits previs a la instal·lació pròpia d'openHAB en la seva versió estable, la que més garanties ofereix, enfront les versió beta i testing:

```

wget -qO - 'https://bintray.com/user/downloadSubjectPublicKey?username=openhab'
| sudo apt-key add -

sudo apt-get install apt-transport-https

echo 'deb https://dl.bintray.com/openhab/apt-repo2 stable main' | sudo tee
/etc/apt/sources.list.d/openhab2.list

```

Així doncs, ja es poden sincronitzar l'índex de paquets i iniciar la instal·lació d'openHAB:

```

sudo apt-get update

sudo apt-get install openhab2

```

A continuació, s'instal·len els paquets d'add-ons per si algun dia interessés desconnectar l'equip de la xarxa i es vol que aquest segueixi funcionant adequadament:

```

sudo apt-get install openhab2-addons

```

Ara ja es pot iniciar OpenHAB i fer que aquest s'iniciï amb el sistema:

```

sudo systemctl start openhab2.service

sudo systemctl status openhab2.service

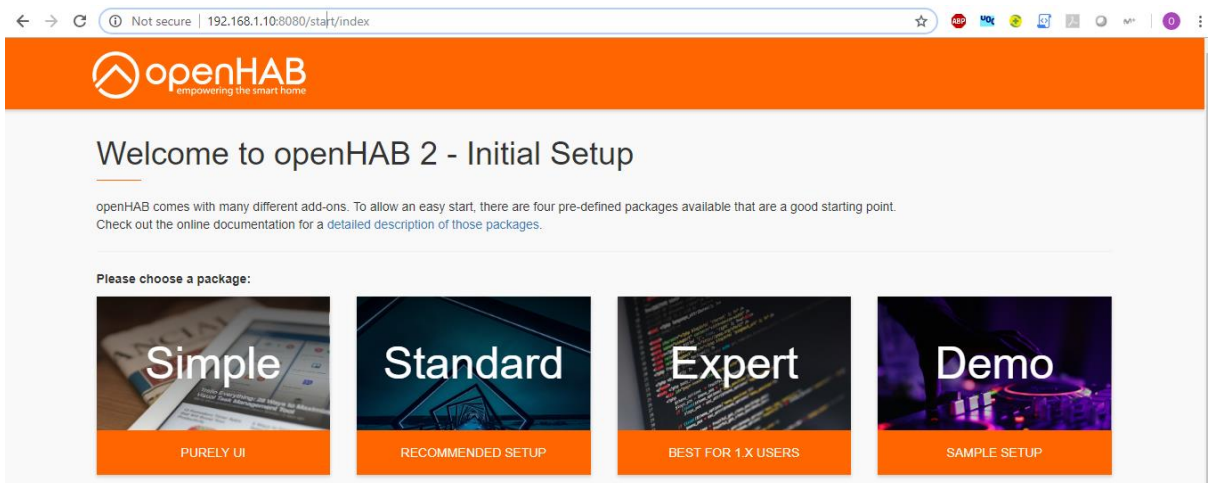
sudo systemctl daemon-reload

sudo systemctl enable openhab2.service

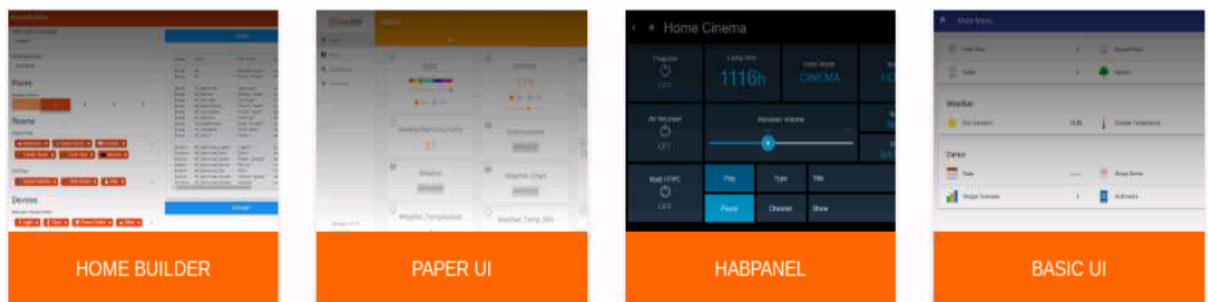
```

I es comprova que està activat i funciona adequadament amb la direcció <http://192.168.1.10:8080>, ja que la instal·lació assigna per defecte el port 8080 per a OpenHAB:

```
● openhab2.service - openHAB 2 - empowering the smart home
Loaded: loaded (/usr/lib/systemd/system/openhab2.service; enabled; vendor pre
Active: active (running) since Thu 2018-10-04 15:18:44 CEST; 50s ago
Docs: https://www.openhab.org/docs/
      https://community.openhab.org
Main PID: 789 (java)
CGroup: /system.slice/openhab2.service
        └─789 /usr/bin/java -Dopenhab.home=/usr/share/openhab2 -Dopenhab.conf
oct 04 15:18:44 raspberry systemd[1]: Started openHAB 2 - empowering the smart h
```



Per últim, es procedeix a instal·lar el paquet estàndard amb la configuració recomanada. En concret, s'instal·len 4 paquets diferents, els quals podran ser modificats, afegir-ne o eliminar-ne segons els interessos de l'usuari:



- Home Builder: ajuda que serveix per a realitzar una primera aproximació als elements que compondran l'estància definida per l'usuari (nom de l'estància, pisos que la componen, nombre d'habitacions...). Així doncs, generarà un 'sitemap' i un 'HABPanel' inicials i assignarà icones clàssiques i grups per als diferents 'items' definits, segons quin sigui el seu tipus.
- Paper UI: des d'aquest menú s'afegiran les diferents 'things' que representaran instàncies físiques del projecte i els diferents 'bindings' per a la seva gestió i control.
- HABPanel: per controlar el sistema amb un dispositiu connectat a la xarxa local. La seva configuració es farà a través de la pròpia interfície o sobre el fitxer amb extensió '.json'.

- Basic UI: mostra els elements configurats sobre els fitxers 'sitemap' i 'items'. A més, presenta la gestió dels diferents 'bindings' instal·lats.

ANNEX IV - Instal·lació broker MQTT: Mosquitto

En aquest apartat es descriu el procediment per instal·lar el 'broker' Mosquitto, que farà les funcions de servidor del protocol MQTT.

Procedim a instal·lar el broker MQTT Mosquitto via SSH (<https://www.instructables.com/id/Installing-MQTT-BrokerMosquitto-on-Raspberry-Pi/>).

S'instal·la el nou repositori:

```
wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key  
sudo apt-key add mosquitto-repo.gpg.key
```

Es permet que aquest estigui disponible per a 'apt' i s'actualitzen paquets:

```
cd /etc/apt/sources.list.d/  
sudo wget http://repo.mosquitto.org/debian/mosquitto-stretch.list  
sudo apt-get update
```

S'afegeix comandes per a fer compatible Mosquitto amb la versió Stretch de Raspbian:

```
cd ~  
wget  
http://ftp.us.debian.org/debian/pool/main/libw/libwebsockets/libwebsockets3_1.2.2-  
1_armhf.deb  
wget http://ftp.us.debian.org/debian/pool/main/o/openssl/libssl1.0.0_1.0.2l-  
1~bpo8+1_armhf.deb  
sudo dpkg -i libwebsockets3_1.2.2-1_armhf.deb  
sudo dpkg -i libssl1.0.0_1.0.2l-1~bpo8+1_armhf.deb
```

I ja es pot instal·lar-lo i actualitzar paquets:

```
sudo apt-get install mosquitto  
sudo apt-get install mosquitto-clients  
sudo apt-get update  
sudo apt-get upgrade
```

Per últim, es comprava que el sistema està actiu:

```
sudo systemctl status mosquitto
```

```
● mosquitto.service - LSB: mosquitto MQTT v3.1 message broker
   Loaded: loaded (/etc/init.d/mosquitto; generated; vendor preset: enabled)
   Active: active (running) since Fri 2018-10-05 20:33:23 CEST; 8min ago
     Docs: man:systemd-sysv-generator(8)
    CGroup: /system.slice/mosquitto.service
           └─5928 /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf
```

I que funciona correctament enviant el missatge “Hola Mosquitto” d'un terminal a l'altre::

```
pi@raspberrypi:~$ mosquitto_sub -d -t prova_mqtt
Client mosqsub|6749-raspberrypi sending CONNECT
Client mosqsub|6749-raspberrypi received CONNACK (0)
Client mosqsub|6749-raspberrypi sending SUBSCRIBE (Mid: 1, Topic: prova_mqtt, QoS: 0)
Client mosqsub|6749-raspberrypi received SUBACK
Subscribed (mid: 1): 0
Client mosqsub|6749-raspberrypi received PUBLISH (d0, q0, r0, m0, 'prova_mqtt', ..
. (14 bytes))
Hola Mosquitto
```

```
pi@raspberrypi:~$ mosquitto_pub -d -t prova_mqtt -m "Hola Mosquitto"
Client mosqpub|6756-raspberrypi sending CONNECT
Client mosqpub|6756-raspberrypi received CONNACK (0)
Client mosqpub|6756-raspberrypi sending PUBLISH (d0, q0, r0, m1, 'prova_mqtt', ..
. (14 bytes))
Client mosqpub|6756-raspberrypi sending DISCONNECT
```


ANNEX V - Instal·lació servidor Samba

Per tal de facilitar l'edició dels fitxers de configuració d'OpenHAB que es troben a la Raspberry, s'instal·la un servidor anomenat Samba i es crea un accés a les diferents carpetes de configuració que interessin: configuració i data d'usuari.

El primer pas serà instal·lar Samba directament (<https://raspberryparatorpes.net/proyectos/instalar-samba-preparando-un-nas-o-servidor-casero-3/>):

```
sudo apt-get install samba samba-common-bin
```

I s'edita el fitxer de configuració per definir les carpetes accessibles des del servidor i facilitar l'edició d'arxius de la Raspberry i per permetre la connexió des de Windows comentant la línia "wins suport = yes" (<https://www.openhab.org/docs/installation/linux.html#recommended-additional-setup-steps>):

```
sudo nano /etc/samba/smb.conf
```

```
[openHAB2-userdata]
comment=openHAB2 userdata
path=/var/lib/openhab2
browseable=Yes
writeable=Yes
only guest=no
public=no
create mask=0777
directory mask=0777
█
[openHAB2-conf]
comment=openHAB2 site configuration
path=/etc/openhab2
browseable=Yes
writeable=Yes
only guest=no
public=no
create mask=0777 directory mask=0777
```

I es crea un usuari per accedir-hi (user: openhab; pass: oscar) i es comprova que es pot accedir a les carpetes mencionades anteriorment:

```
sudo smbpasswd -a openhab
```

```
sudo chown -hR openhab:openhab /etc/openhab2
```

I es reinicia el servidor:

```
sudo systemctl restart smbd.service
```

Finalment, es connecta a les dues carpetes mencionades:

← Map Network Drive

What network folder would you like to map?

Specify the drive letter for the connection and the folder that you want to connect to:

Drive: Z:

Folder: \\192.168.1.10\openh2b2-conf

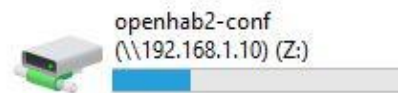
Example: \\server\share

Reconnect at sign-in

Connect using different credentials

[Connect to a Web site that you can use to store your documents and pictures.](#)

I es comprova que ja estan situades dins les unitats de xarxa compartides:



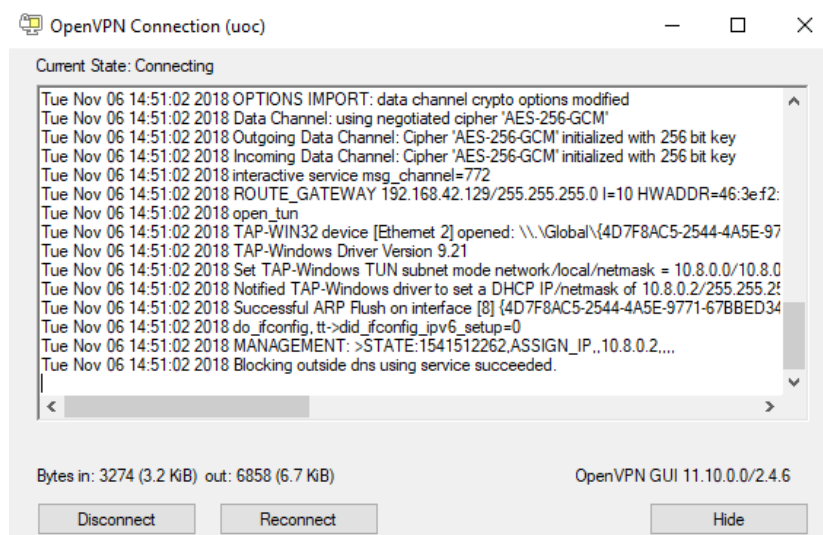
ANNEX VI - Instal·lació addicional

VI.I - OpenVPN-GUI [23]

Per tal de poder accedir al sistema via VPN privada des de l'ordinador personal i qualsevol lloc del món, es carrega el paquet que inclou software i interfície gràfica OpenVPN-GUI (<https://openvpn.net/index.php/open-source/downloads.html>) i s'instal·len a l'ordinador portàtil.



Un cop descarregat i instal·lat, ja es pot accedir amb l'usuari creat anteriorment a PiVPN:



VI.II - Notepad++



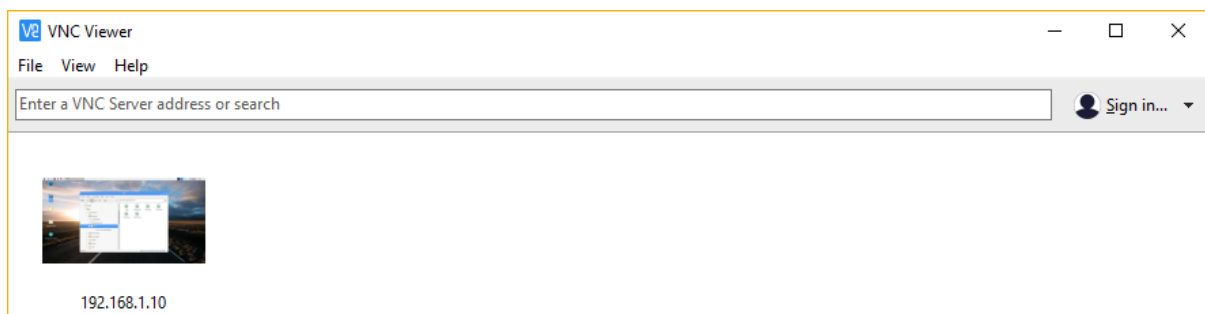
Per tal de poder editar de manera còmode i en un entorn de treball més amè que el que proporciona un terminal de Raspbian els fitxers de configuració d'OpenHAB, s'instal·la l'editor de text Notepad++ [20] a l'ordinador des del qual es connectarà a les unitats de xarxa compartides anteriorment.

A més, s'instal·len les llibreries corresponents als diferents tipus d'objectes (items, rules...) d'OpenHAB per tal que, de manera visual, es pugui comprovar que la sintaxi del llenguatge és la correcta i que no hi ha cap error al codi (<https://github.com/thefrip/openhab-samples>):

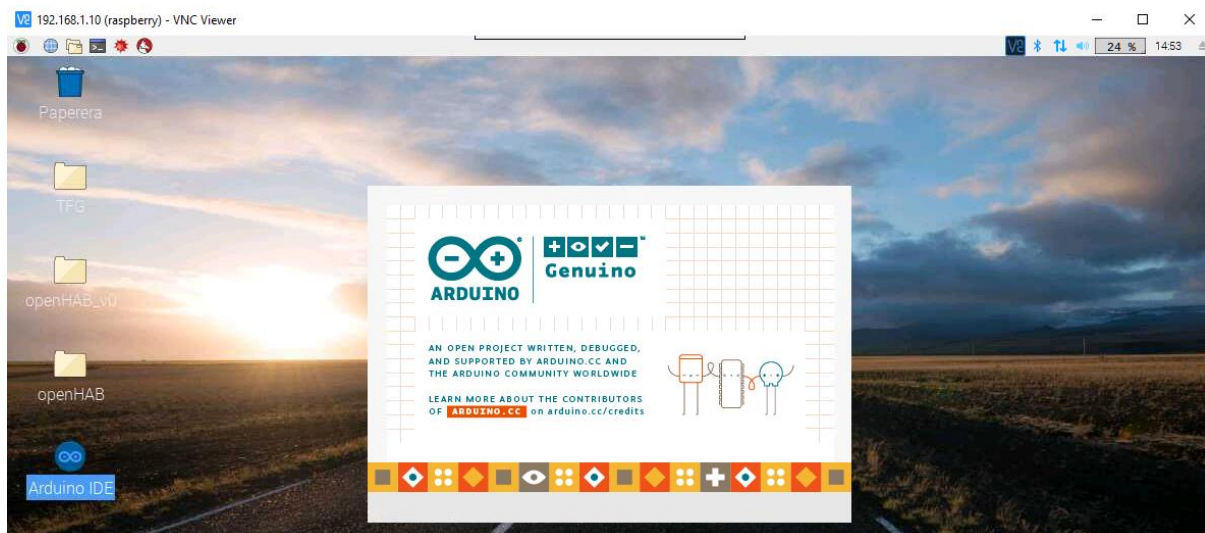
openHAB-Items.xml	Files uploaded from Google Group openhab	3 years ago
openHAB-Persist.xml	Files uploaded from Google Group openhab	3 years ago
openHAB-Rules.xml	Files uploaded from Google Group openhab	3 years ago
openHAB-Sitemap.xml	Files uploaded from Google Group openhab	3 years ago

VI.III - VNC Viewer [21]

Un cop creat el client OpenVPN, s'instal·la en l'ordinador personal VNC Viewer, un software que permet accedir a l'interfície gràfica del dispositiu Raspberry un cop a la xarxa local:



Així doncs, amb aquest software, un cop introduïda la IP, es podrà accedir de manera remota al seu escriptori i, mitjançant l'IDE d'Arduino, un cop aquest es trobi connectat via USB, es procedirà a la seva programació:



VI.IV - NoIP

Per tal de sortir a la xarxa a través del proveïdor de serveis de Internet (ISP), aquest assigna una adreça IP pública que va canviant amb el temps [18]. Un problema que presenta aquest fet, és que s'hauria de canviar, cada cop que aquesta adreça esdevingui diferent, l'adreça a través de la qual s'accedeix a la VPN creada. Així doncs, i tal com s'ha configurat en la instal·lació de PiVPN, s'accedeix a la xarxa a través d'un domini i no a través d'una adreça IP. Aquest domini, es crearà dins la pàgina <https://www.noip.com/>, la qual ofereix un servei de hosts gratuïts.

Per últim, cal que aquest domini estigui referenciat sempre a la IP pública d'aquell determinat moment, per això, es configura el router per tal que aquest actualitzi la direcció IP assignada al domini creat amb l'usuari de NoIP. Aquest pas es realitza dins la pròpia configuració del router, aprofitant el servei Dynamic DNS, de la manera següent:

configuración del servidor de nombres dinamico (DDNS)

servicio	nombre de host completo	nombre de usuario email	contraseña	última actualización	
No-IP ▼	<input type="text"/>	<input type="text"/>	<input type="text"/>		<input type="button" value="añadir"/>
No-IP	<input type="text"/>	<input type="text"/>	*****	05/10/18 11:52:55	<input type="button" value="borrar"/>

ANNEX VII - Instal·lació Camera Suite per a GoPRO Hero4

Per tal de poder accedir remotament a l'enregistrament de la càmera GoPRO Hero 4, s'instal·la el software Camera Suite al dispositiu Raspberry Pi.

Per a fer-ho, s'executen les següents comandes a un terminal:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

```
sudo apt-get install qt5-default libqt5script5
```

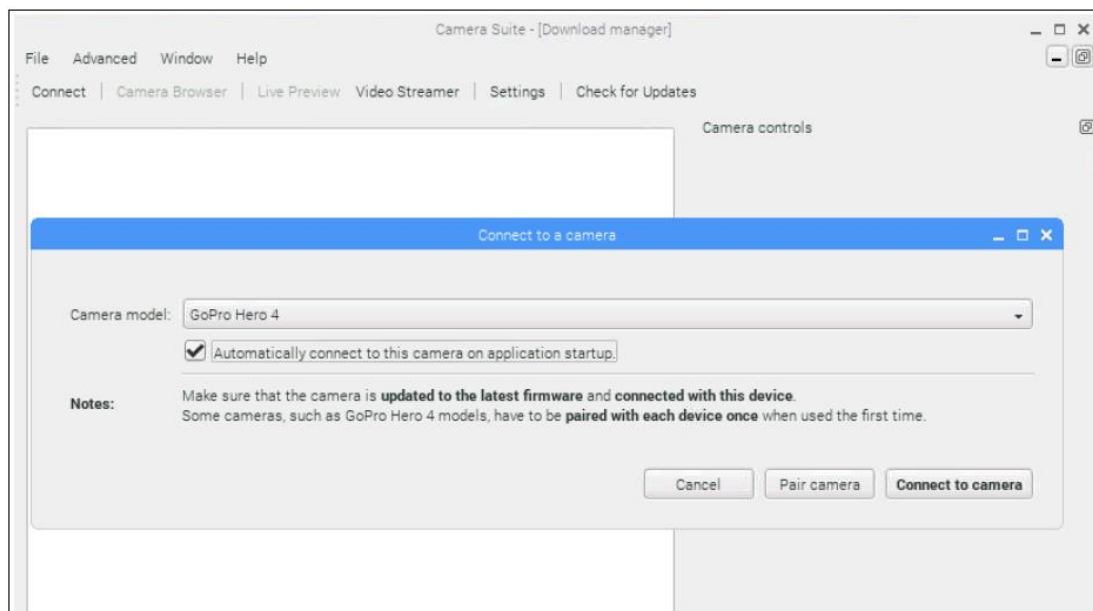
Un cop executades les 3 anteriors comandes, s'accedeix al directori on s'ha descarregat prèviament el software i es descomprimeix, per últim, s'executa l'arxiu:

```
cd /home/pi/Desktop
```

```
tar xvf camerasuitepi.tar.gz
```

```
./camerasuite
```

Ara ja tan sols resta configurar la càmera:



I es comprova com s'obté la imatge en directe a través del 'live preview' del propi programa per monitoritzar la maqueta telemàticament:

