

**Desenvolupament de Control domòtic de
Pàrquings amb plataforma Arduino.**

David Flores Galí



*Desenvolupament de Control domòtic de Pàrquings amb
plataforma Arduino.*

David Flores

Pla d'estudis de l'estudiant

Àrea de treball final

Consultor/a: Oriol Jaumandreu Sellarés

Professor/a responsable de l'assignatura: Pere Tuset Peiró

15-01-2017



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Copyright ©David Flores Galí.

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Desenvolupament de Control domòtic de Pàrquings amb plataforma Arduino.</i>
Nom de l'autor:	<i>David Flores Galí</i>
Nom del consultor/a:	<i>Oriol Jaumandreu Sellarés</i>
Nom del PRA:	<i>Pere Tuset Peiró</i>
Data de lliurament (mm/aaaa):	<i>01/2017</i>
Titulació o programa:	<i>Grau d'Enginyeria Informàtica</i>
Àrea del Treball Final:	<i>Arduino.</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>Pàrquing, domòtica, Arduino</i>
Resum del Treball (màxim 250 paraules): <i>Amb la finalitat, context d'aplicació, metodologia, resultats i conclusions del treball</i>	
<p>La finalitat d'aquest treball rau a dotar d'una gestió remota a un pàrquing situat a 100 Km de les oficines que gestionen aquest immoble.</p> <p>Des de les oficines es vol controlar les alarmes d'incendi, i en cas que es dispari el pressòstat de l'aigua. Així mateix volen poder controlar l'obertura de portes, tant la porta principal de vehicles com la de vianants, en cas que hi hagi alguna visita per llogar una plaça d'aparcament.</p> <p>La solució proposada es basa en un sistema gestionar per una placa Arduino amb connectivitat a Internet que permetrà via web gestionar si es produeix una incidència mitjançant els sistemes d'entrada d'Arduino i amb un botó a la web efectuar accions com obertura de portes, tancament d'alarma o gir de control de pressòstat o ventilació Per a executar aquestes funcions, caldrà un sistema Arduino amb connectivitat Ethernet que permetrà connectar-se a Internet per crear un portal web on l'usuari pugui interactuar i enviar mitjançant SMTP correus electrònics per avisar d'incidències.</p> <p>Aquest sistema a més és escalable de manera que es pot afegir sensors al sistema Arduino que permetin gestionar des d'Humitat a nivells de CO i sortides per realitzar les accions pertinents. Ara bé depenent del nombre d'entrades i sortides que es requereixi caldrà adquirir un model més o menys avançat d'arduino que permeti.</p>	

Abstract (in English, 250 words or less):

The purpose of this work lies in a remote management equips a parking located 100 km from the offices that manage this property.

We want to control fire alarms and if they shoot the water pressure switch from the office. Also we want to control the opening of doors (both the front door of vehicles and pedestrian, if there is any visit to rent a parking space).

The proposed solution is based on a system managed by an Arduino board with Internet connectivity that allow management with a webpage if an incident occurs through the system input Arduino and some buttons on the website to perform actions such as opening doors, closing alarm or turn control ventilation or pressure switch

To perform these functions, you need a system with Arduino connectivity

Ethernet which will connect to the internet to create a website where users can interact and send emails via SMTP to warn of incidents.

This system is also scalable so that you can add sensors that allow the Arduino from managing humidity levels of CO and exits to perform the appropriate actions. However, depending on the number of inputs and outputs that are needed will acquire a more or less advanced Arduino how to manage the number of inputs and outputs required.

Índex

1. Introducció.....	6
1.1 Context i justificació del Treball	6
1.2 Objectius del Treball.....	7
1.3 Hipotesi del projecte	8
1.4 Enfocament i mètode seguit	9
1.5 Planificació del Treball.....	10
1.6 Breu sumari de productes obtinguts	12
1.7 Breu descripció dels altres capítols de la memòria	12
2. Plataforma de desenvolupament.....	14
2.0 Estat anterior del projecte.....	14
2.1 Requisits del sistema.....	16
2.2 Hardware necessari	17
2.3 Entorn de programació.....	21
2.4 Disseny del projecte:	27
2.5 Problemes detectats:.....	44
2.6.Valoració econòmica	46
3. Conclusions.....	47
4. Glossari	48
5. Bibliografia.....	49
6. Annexos	50
6.1 Diagrama d'estats.....	50
6.2 Codi Font:.....	52

1. Introducció

1.1 Context i justificació del Treball

L'increment dels dispositius que estan interconnectats entre si fa que cada vegada apareguin més objectes que treballen sense la interacció humana: des d'una nevera que pot mesurar la quantitat de productes disponibles i realitzar una comanda cada vegada que algun element estigui a punt d'esgotar-se, fins a places d'aparcament que es comuniquen amb el nostre vehicle quan estan disponibles. Aquesta és la revolució de la Internet de les coses o "Internet of Things".

Bàsicament cal disposar de diferents dispositius que tinguin connexió a internet, sensors per mesurar el que es vulgui controlar i un protocol comú que permeti que s'estableixi una comunicació entre els aparells interconnectats. Es podria assumir que s'inclourien en la definició de dispositius d'Internet of Things, els sistemes domòtics que gestionen diferents aparells del domicili en funció de certes variables: com pot ser el tancament d'una persiana en cas que s'arribi a un llindar d'un sensor de llum o que l'usuari pugui des del telèfon mòbil engegar la caldera.

En aquest projecte es voldrà fer una aproximació a "Internet of Things". I més concret en certs aspectes de la domòtica, on segons variïn les condicions d'una ubicació, l'usuari pugui rebre notificacions o bé actuar en certs dispositius per mitjà d'una connexió a internet. En concret es vol realitzar la creació d'un sistema basat en Arduino que permeti a uns usuaris que treballen en unes oficines situades a la província de Girona conèixer l'estat a temps real de diferents elements d'un pàrquing ubicat a Barcelona.

Inicialment aquesta ubicació disposa d'un sistema de control de portes connectat a una targeta de relés de quatre ports de sortida gestionada pel sistema de càmeres GEOVISION GV1420 i quatre ports d'entrada que permetrien controlar sistemes d'entrada digital com podria ser el sistema de foc i d'aigua, Però les limitacions del sistema només permetés connectar els avisos via Email o SMS a una sola porta d'entrada, de manera que només es podria controlar el sistema de foc o el d'aigua o bé connectant les dues variables en la mateixa entrada de manera que informes que hi ha una alarma però sense distingir quin element és el causant.

En el cas que l'ordinador no respongués, es perdria l'accés a controlar el sistema de portes i calia enviar a algú de manteniment per reiniciar la maquina. De manera que per evitar aquesta dependència caldria disposar d'un sistema aliè a l'ordinador que no es vegi afectat en cas de caiguda i que a més pugui permetre interactuar amb el sistema informàtic que controla les càmeres per poder apagar-lo o iniciar-lo a conveniència.

Actualment en el mercat es disposa de múltiples sistemes domòtics que climatització, racionalització energètica, sensorialització de variables analògiques o control d'alertes de determinats paràmetres. Fins i tot múltiples empreses gestionen projectes a mesura que poden realitzar aquestes tasques. Però la base d'aquest projecte rau a implementar un sistema de baix cost i que permeti una introducció a la realització de sistemes de domòtica de cara a la implantació de projectes de major envergadura i el sistema Arduino és una bona eina de cara a la introducció de projectes electrònics amb microcontroladors.

1.2 Objectius del Treball

Objectius Principals:

Els objectius principals que es pretenen amb aquest projecte són:

- Establir un sistema de control d'estats de determinats sistemes d'un pàrquing, com és l'estat de l'alarma de focs, pressòstat i paràmetres ambientals
- Implementar un sistema que permeti a l'usuari interactuar amb sistemes remots per activar o desactivar determinats dispositius. Permetent engegar o apagar un ordinador, engegar o apagar la ventilació i obrir les portes de vehicles i de vianants
- Fer que la comunicació es realitzi mitjançant una interfície web accessible des de qualsevol part del món.
- Emprar un sistema d'open hardware, amb el que no hi ha restriccions per part del fabricant i s'obté millor coneixement dels elements que s'utilitzen.
- Emprar un sistema de Software lliure pel desenvolupament emprant el IDE subministrat per Arduino i que permet una reducció del cost perquè no cal adquirir llicències per a poder desenvolupar amb aquesta plataforma.

Objectius Secundaris:

A part dels objectius primaris també es pretendrà donar resposta als següents objectius:

- Utilitzar els coneixements que s'han adquirit durant el grau per desenvolupar un projecte que pugui ser aplicable a un cas determinat i tingui una utilitat real
- Adquirir coneixements de desenvolupament de la plataforma Arduino
- Començar a familiaritzar-se amb entorns orientats a la domòtica i a l'Internet de les Coses.

1.3 Hipòtesi del projecte

S'espera del projecte obtenir un producte que permeti en concret realitzar el control d'estats digitals i analògics i permetre un control de sortides mitjançant una pàgina web com a interfície entre l'usuari i el dispositiu.

Els elements que caldrà controlar per l'usuari seran l'estat de l'alarma de foc (si està o no sonant), l'estat de la pressió de la instal·lació d'aigua (a fi de determinar si hi ha o no pèrdues de pressió) i la concentració de Monòxid de Carboni per determinar si cal o no activar el sistema de ventilació. En cas que un d'aquest element deixi d'estar en el seu estat correcte, es realitzaran una notificació als usuaris en forma de correu electrònic que s'anirà repetint periòdicament fins que la variable torni al seu estat normal.

La part que es vol que l'usuari pugui controlar remotament serà l'obertura de portes de vehicles i de vianants (a fi que permetin el pas en cas que vingui un possible client a fer una visita per a poder llogar un aparcament), engegar i apagar de l'ordinador que controla el sistema de càmeres i permetre el control del funcionament del sistema de ventilació quan la concentració de Monòxid de Carboni sigui la correcta.

De manera que per poder donar com a vàlid el producte caldrà que realitzi les següents funcions:

- Captura de valor digital pel control del pressòstat del sistema de canalització d'aigües.
- Captura de valor digital que indica que l'alarma està sonant.
- Captura mitjançant un sensor analògic de la superació d'un llindar respecte a la concentració de Monòxid de Carboni.
- En cas que s'activi l'avís de monòxid de Carboni activa un relé que engegui el sistema de ventilació.

- Quan s'activi una de controls digitals o analògics cal que s'envii un correu electrònic notificant de la incidència i que es repeteixi amb una freqüència de 3 minuts.
- Permeti l'activació d'un relé per engegar o apagar un ordinador
- Permeti l'activació d'un relé per activar el motor de la porta de vehicles
- Permeti l'activació d'un relé per obrir la porta de vianants
- Que el dispositiu generi una pàgina web que mostri els estats de les entrades digitals i analògiques i permeti l'activació dels relés.

1.4 Enfocament i mètode seguit

Per realitzar el projecte de disseny d'un sistema basat amb: la tecnologia Arduino caldrà efectuar un elevat nombre de tasques de forma seqüencial (una tasca es farà després d'un altre) i en alguns casos es desenvoluparan de forma paral·lela (en un mateix període s'efectuaran diverses tasques que no tindran relació de dependència entre elles).

Caldrà per a la raó esmentada crear un calendari on s'especifiqui les tasques que cal desenvolupar i el període en què hagi d'estar completades de forma més realista possible.

En primer lloc s'elaborarà un diagrama de Gantt que s'especifiqui les diferents tasques que calgui concloure i el temps necessari per a poder assolir les fites i en cas que hi hagi alguna desviació important caldrà revisar el diagrama per adaptar-lo reajustant les diferents tasques a fi de complir amb les dates previstes en l'entrega de les diferents PAC de l'assignatura.

Entre les diferents tasques hi haurà l'obtenció de documentació, s'elaborarà un estudi de necessitats de disseny i material per assolir les fites requerides segons els objectius definits en el projecte.

Posteriorment es farà la compra del material, tant els mòduls d'Arduino com els Shields necessaris i el cables que calgui per a poder establir les diferents connexions i dels sensors que s'utilitzarà per a l'anàlisi ambiental.

Més endavant i una vegada es tingui com a mínim la placa Arduino adquirida, es faran de proves de programació per a entrar en contacte amb el llenguatge i l'entorn d'Arduino com són el típic "Hello World" que en aquest cas serà fer que un LED faci pampallugues en una freqüència determinada i posteriorment es desenvoluparan altres programes per familiaritzar-se i aprofundir amb els mòduls necessaris en el projecte com serà la comunicació per port sèrie (per obtenir els diferents logs interns d'execució de programa), la connexió en xarxa i la creació dinàmica d'una pàgina web. Més endavant es realitzaran proves d'activació de relés i funcionament del sensor analògic i l'enviament de correu electrònic.

Finalment amb el codi generat anteriorment es podrà elaborar una codificació que englobi totes les funcions requerides: Connexió xarxa, creació web, control d'entrades de pins digitals i analògics (en aquest cas s'establirà un valor que faci de llindar aleatòriament per a la realització de proves), activació de relés i enviament de correu electrònic.

Després s'elaborarà els diferents prototips i esquemes per entregar en la memòria de la pràctica. Es prioritzarà la part funcional tant en el disseny de Hardware com en la pàgina web d'interacció amb l'usuari, preferint en cas de no disposar de prou temps d'assolir totes les funcionalitats enfront d'un disseny elegant a la vista.

Finalment, i si es disposa de prou temps, s'intentarà fer que el disseny de la pàgina web sigui prou elegant i agradable a la vista i s'estudiarà com calibrar el sistema analògic per a poder traduir el valor de voltatge obtingut pel sensor en una variable física com pot ser les Parts Per Milió (PPM).

A finals de desembre o principis de gener es preveu efectuar una visita a la ubicació de l'aparcament per a executar la instal·lació definitiva.

Caldrà en la part de disseny i en cas de preveure problemes intentar augmentar la comunicació amb el coordinador i si cal proposar conferències via Skype si fos necessari.

El projecte es dividirà en les següents fases:

1. Investigació i documentació d'Arduino: En aquesta fase es farà una cerca de documentació a fi de veure quines possibilitats permet la plataforma de Hardware lliure Arduino.

2. Determinar elements de Hardware necessaris: En aquesta fase que està lligada en l'anterior, es buscarà quins mòduls (plaques i Shields d'Arduino) caldrà adquirir per a l'acompliment dels requisits per a poder resoldre els problemes plantejats en la tesi del projecte. Així com els sensors o controls addicionals per a la plataforma que es vol desenvolupar. I finalment fer la compra del material necessari

3. Aprenentatge Arduino: Una vegada adquirit el material s'entrarà en la fase de presa de contacte amb la instal·lació de l'IDE i la realització de les primeres proves per determinar el funcionament i com programar la plataforma. També s'estudiarà el sistema de programació (com s'estructura un programa i la seva sintaxi).

4. Crear prototips i programació de tasques individuals: Aplicant el mètode de "dividir i vèncer", es parteix el problema en parts petites i es desenvolupa en l'àmbit de hardware i software un prototip per una part del problema (obertura de portes, alarma de focs, etc.), connexió Ethernet i creació de pàgines webs dinàmiques. Aquesta fase es dividirà en els següents apartats:

4.1. Connexió TCP/IP i pàgines Webs dinàmiques.

4.2 Control d'entrada digital (Foc i Aigua) i mostrar sortida per pàgina web.

4.3. Activació de relés per mitjà de botonera en pàgina web.

4.4 Control d'estat de sensor de CO i mostrar sortida en Pàgina Web.

4.5 Enviament de correu electrònic per activació per mitjà de botons en pàgina web.

5. Prototip definitiu i programació: Adjuntar tota la part de codi generat en el pas 4 i integrar-ho correctament i programar la part de la generació

dinàmica del HTML segons les variables a partir del codi estàtic preparat en el pas 5 i fer proves de validació. En aquest apartat es documentarà amb un vídeo que s'entregarà com a part de la PAC2.

6. Preparació de la connexió definitiva en l'Aparcament, caldrà refinar el codi, coordinar-se amb personal de manteniment per la tirada de cable i fer una visita per fer la instal·lació.

7. Redacció del projecte: Aquesta fase inclou la documentació de la memòria en els diferents passos del projecte.

1.6 Breu sumari de productes obtinguts

Com a producte obtingut es disposa d'un sistema basat en Arduino que permet el control de 2 entrades digitals (aigua i foc) i una d'analògica. Aquesta placa està connectada amb un Shield de 4 relés que permet l'actuació d'engegada o aturada de 4 dispositius (motor porta de Vehicles, sistema d'obertura de porta de vianants, ventilació i interruptor de motor).

El sistema també permet l'accés a través d'Internet a una pàgina web generada pel dispositiu i enviar un correu electrònic com a notificació en cas que s'activi una entrada digital o una analògica.

1.7 Breu descripció dels altres capítols de la memòria

En el següent capítol es descriurà les diferents parts del desenvolupament del projecte, des de l'estat anterior que hi ha a la ubicació on s'aplicarà el projecte. Més endavant s'analitzarà els diferents necessitats que ha de cobrir el projecte i quin serà el maquinari que caldrà per a realitzar-ho. Després en el mateix capítol es comentarà quin és l'entorn del programari necessari per desenvolupar l'aplicació. També hi ha un subapartat on es documenta com s'han resolt els diferents elements del disseny del producte tant en l'àmbit de maquinari (des de la descripció dels ports d'entrada i sortida que s'utilitzaran com els sensors), Com es definirà la pàgina web que crearà el dispositiu i de quina manera s'enviaran les notificacions. Finalment en el segon capítol

s'explicaran quin son els principals problemes que s'ha trobat durant la creació del projecte i quin és el cost del material emprat.

Els capítols següents seran per a la descripció de les conclusions, el glossari de les paraules més emprades en la memòria i la bibliografia.

Com a annexos s'afegirà els diferents diagrames d'estats del programa i el darrer codi font.

2. Plataforma de desenvolupament

En aquest capítol es descriurà quins han estat les plataformes utilitzades per a la realització del projecte, detallant les necessitats del projecte segons els requisits esmentats en capítols anteriors, així com la determinació del maquinari i programari que caldrà emprar per a poder desenvolupar la solució al problema que s'ha determinat esmenar.

2.0 Estat anterior del projecte

En l'edifici d'aparcaments on s'ha d'ubicar el sistema a desenvolupar en aquest projecte, hi ha actualment connectat un sistema similar de menors prestacions que serà substituït pel que s'estan dissenyant.

El sistema actual disposa de la connexió dels relés de la porta a una placa GV-IO NET de Geovision. Aquesta targeta està connectada a una altra targeta captadora d'imatge pel sistema de càmeres de l'aparcament. Des del programari de control de càmeres es pot accedir al sistema de control d'entrades i sortides i es poden realitzar les següents funcions: Activar el relé de la porta de vehicles

- Activar el relé de la porta de vehicles
- Control de senyal digital de focs



Figura 2 Tarja GV-IO NET de Geovision.

Specifications			
OS Supported	32-bit	Windows XP / Vista / 7 / 8 / 8.1 / 10 / Server 2008	
	64-bit	Windows 7 / 8 / 8.1 / 10 / Server 2008 / Server 2012 / Server 2012 R2	
Input	Input	4	
	Input Signal	Dry-Contact	
		Wet-Contact : 9~30V AC/DC	
Output	Relay Output	4	
	Relay Status	Normal Open	
	Relay Capacitance	USB Connection	30V DC, 3A
		RS-232 Connection	125 / 250V AC, 3A 30V DC, 3A
Interface		RJ-11 to DB9	
		RJ-11 to USB	
		3-pin internal USB to internal USB	
Mode Switch	I/O Box Mode	Without GV-Video Capture Card	
	NET/IO Card Mode	With GV-Video Capture Card	
Address		1-4	
Communication		RS-485, USB, RS-232	
Environmental Conditions		0~50°C / 32~122°F, 5%~95% (non-condensing)	
Dimensions(W x H)		90 x 99 (mm) / 3.90 x 3.54 (in)	

Figura 3 Especificacions tarja GV-IO NET de Geovision.

En aquest sistema, no s'ha implementat el control d'aigua ni el de gasos, però per les proves realitzades en sistema de Geovisión només es pot assignar l'enviament de notificacions només en una porta d'entrada de manera que no seria viable a priori establir un sistema de notificacions per les diferents alarmes.

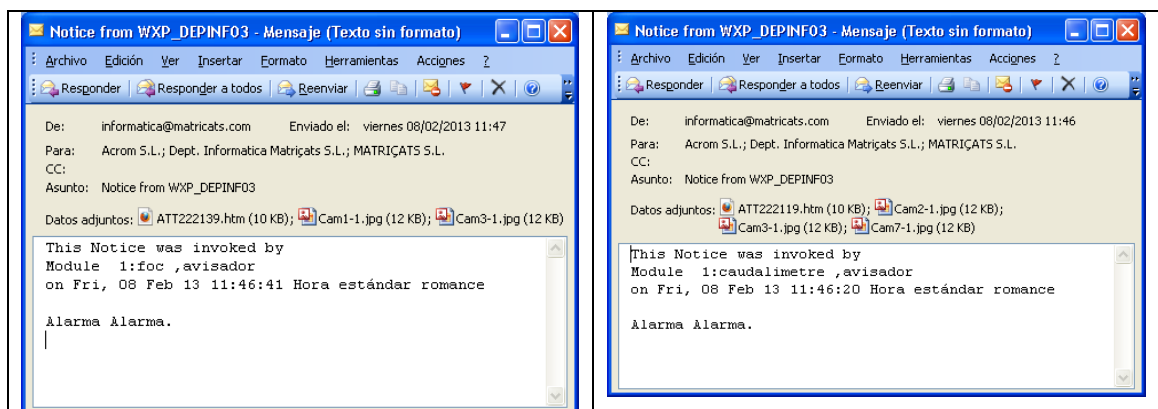


Figura 4 Captures de pantalla de notificació de focs

En principi hi ha assignat a l'alarma de focs l'enviament de notificacions. I en cas que es volgués enviar una notificació per més d'un senyal, com per exemple la de control de pressòstat, caldria establir una entrada que fos comuna a la resta d'entrades. Això es faria desviant l'entrada de focs i aigües a una tercera entrada. D'aquesta manera quan s'actives l'entrada de focs i/o la d'aigües també s'activaria la tercera (notificacions) que enviaria un correu quan s'ha activat una senyal. Perquè no és derives corrent de foc quan s'actives a la d'aigües per mitjà del senyal de notificació es connectarien uns díodes per

evitar la pujada de corrent aigües amunt, com podrà apreciar-se en la figura següent.

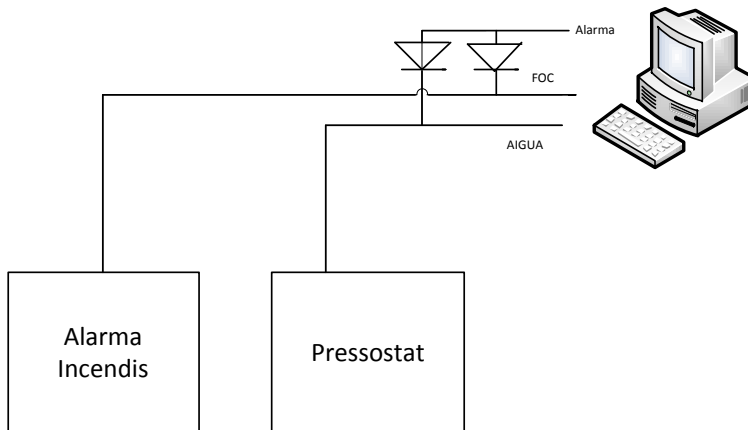


Figura 5 Diagrama de connexió d'entrades digitals en tarja GV-IO NET

A més aquest sistema no incorpora pins d'entrada/sortida analògics de manera que no seria viable l'obtenció de dades com el nivell de gasos, temperatura o humitats per posar un exemple.

Aquesta targeta controladora (GV 1240) que es requereix per a afegir un altre d'entrada/sortida té un cost actual de 889€ i la targeta amb mòduls d'entrada/sortida (GV-IO net amb 4 ports d'entrada i 4 de sortida) digital té un cost 79€.

A més aquest sistema està instal·lat en un ordinador de manera que el funcionament del control d'entrada i sortida depèn de si l'ordinador està engegat i que el programa de la controladora estigui actiu.

2.1 Requisits del sistema

La idea del TFG és implementar un sistema que controli un seguit de senyals d'entrada per determinar si hi ha algun canvi en els sistemes d'alarma, pressostat i control d'aire. També haurà de permetre l'activació d'uns motors per mitja de relés. I aquest estats o activacions del sistema haurà de poder-se consultar o activar a través d'una pàgina web.

D'aquesta manera ens caldrà una plataforma d'Hardware que tingui aquests requisits:

- Entrades:

- Control Alarma de focs
- Control canvi de pressió aigua
- Nivell de Monòxid de Carboni

- Sortides:
 - Porta de vianants
 - Porta de Vehicles
 - Ventilació

- Comunicacions:
 - Permetre creació pàgina web dinàmica
 - Accés TCP/IP
 - Permetre enviament de correus electrònic en cas d'avisos

2.2 Hardware necessari

A fi de poder satisfer les necessitats esmentades en l'apartat anterior caldrà una placa que permeti 3 entrades, 3 sortides i comunicació TCP/IP.

En aquest apartat es descriurà diferents alternatives que podrien emprar-se i finalment quin és el material amb què ens quedem.

Placa base Arduino:

En aquest apartat s'ha plantejat el dilema d'adquirir un model bàsic com Arduino UNO o superior i adquirir per realitzar les tasques de comunicació un Shield Ethernet o Shield WiFi o bé adquirir una placa Arduino Ethernet i tenir en un sol component l'apartat base i comunicacions cobert.

Arduino UNO + Arduino Ethernet Shield

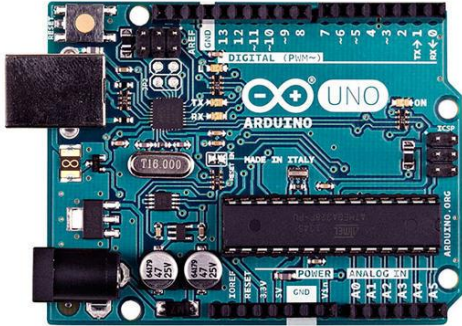


Figura 6 Model Arduino UNO

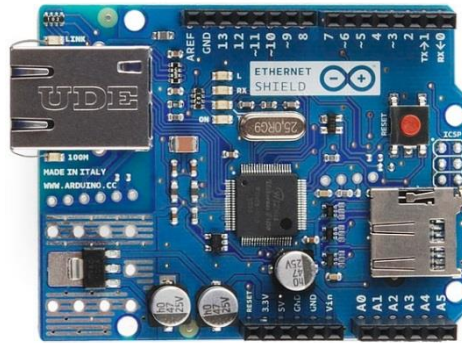


Figura 7 Shield Arduino Ethernet

El model Arduino UNO té un cost de 19,95€ i l'Arduino Ethernet Shield v2 de 22,90€

Arduino Shield

Aquest dispositiu incorpora les funcionalitats que disposa Arduino UNO, en pins digitals i analògics però porta incorporat la funcionalitat de connexió per Ethernet. De manera que equivaldria a disposar d'un model Arduino UNO més un Shield Arduino Ethernet en una sola placa. Aquest model té un cost de 42,90€.



Figura 8 Model Arduino Ethernet

L'Arduino Ethernet Com a característiques té

Micro controlador	ATmega328
Voltatge Operació	5V
Voltatge d'entrada Recomanat	7-12V
Pins digitals E/S	14 , 4 proveeixen sortida PWM
Pins d'entrada analògica	6
Pins Reservats	10 de 13 utilitzats per SPI 4 utilitzats per SD card
Consum per pin E/S	40 mA
Consum del pin 3.3V	50 mA
Memòria flash	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Freqüència de rellotge	16 MHz

Taula 1 característiques de Arduino Ethernet

Finalment s'ha optat per adquirir la placa Arduino Ethernet a fi que el sistema no adquireixi un volum massa elevat ja que caldrà adquirir més Shields per altres funcions, a més la diferencia de cost entre el pack Arduino UNO + Ethernet Shield i Arduino Ethernet és de 0,05 €. En cas que la diferencia hagués estat notable possiblement s' hauria decantat per l'opció de la placa bàsica més el Shield .

Arduino 4 relay Shield

Aquest dispositiu permet l'activació de 4 relés. Aquests relés es controlen per mitjà dels pins 4,5,6 i 7 de la placa Arduino.

D'aquesta manera des d'Arduino es pot encendre o apagar fins a 4 dispositius que estiguin connectats als relés d'aquesta placa.



Figura 9 Arduino Relay Shield

Arduino Screw Shield

No aporta cap funcionalitat pel que fa a maquinari addicional però ens permet connectar d'una forma més definitiva els cables del sistema cap als diferents pins analògics o digitals de la placa Arduino amb connectors de cargol de 3.5 mm. I també permet continuar intercalant altres plaques Shield sobre seu.

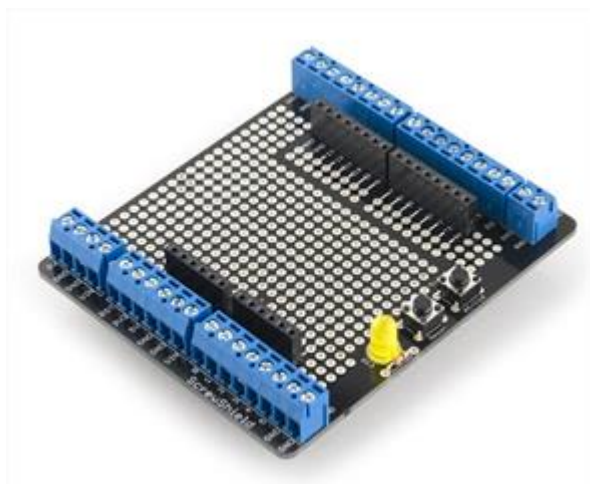


Figura 10 Arduino Screw Shield

Sensor analògic de Monòxid de carboni

El mòdul del “gas sensor v 1.3” conté una placa amb les amb el sensor MQ7 més el circuit on es connecta el sensor per facilitat la tasca de connexió amb plaques com Arduino. Incorpora 4 pins dels quals són:

- Pin de connexió a 5 v
- Pin de terra
- Pin de sortida Digital
- Pin de sortida Analògica



Figura 11 gas sensor v.1.3

2.3 Entorn de programació

2.3.1 IDE Arduino

Malgrat que existeix mecanismes com un editor de text i un programador de microcontrolador o amb IDE com Eclipse o Netbeans amb els seus "plugins" per Arduino, per a poder desenvolupar la programació, compilació i enviament de la programació a l'Arduino. S'ha optat per fer-

ho mitjançant el programari gratuït disponible de la pàgina oficial, la versió actual és la 1.6.12.



Figura 12 Captura de pantalla de IDE oficial d'Arduino amb el codi d'exemple Blink

En aquesta IDE, el programa o "Sketch" es realitzarà en el seu editor de text i posteriorment es compilarà i s'enviarà per port sèrie que tingui assignat l'Arduino a la memòria del microcontrolador.

Aquesta IDE també eina per facilitar la tasca de programació com accés a llibreries, detecció de ports i de dispositius i detecció d'errors i també disposa de l'opció de visualitzar establir comunicació entre l'ordinador i el dispositiu per mitjà del port sèrie amb un monitor en què es pot veure missatges generats cap a port sèrie del dispositiu o bé comunicar-se des de l'ordinador cap al dispositiu amb el mateix port.

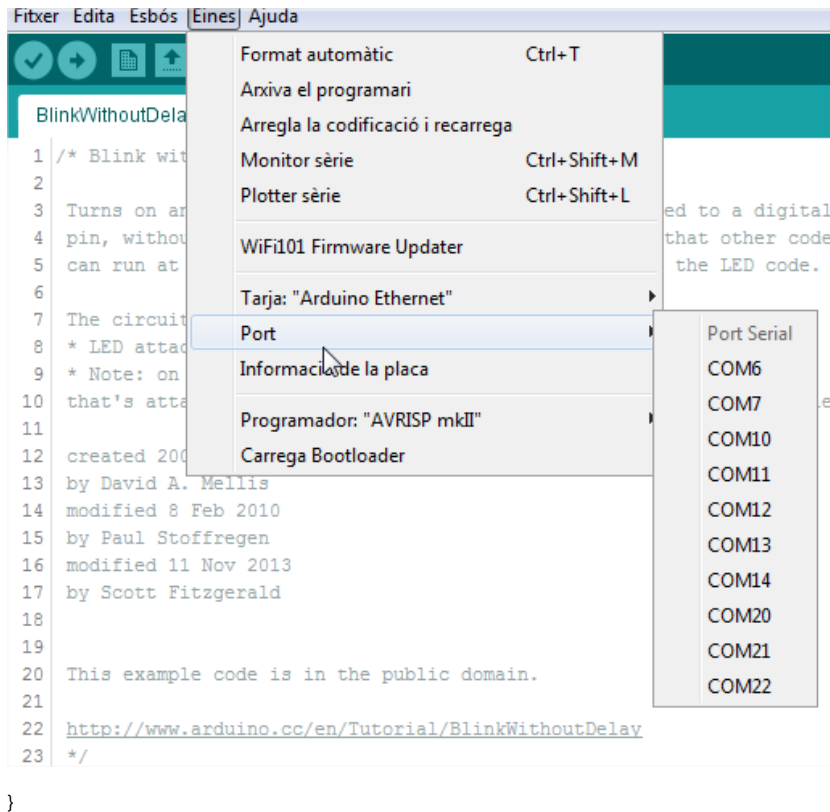


Figura 13 selecció de ports des del IDE de Arduino

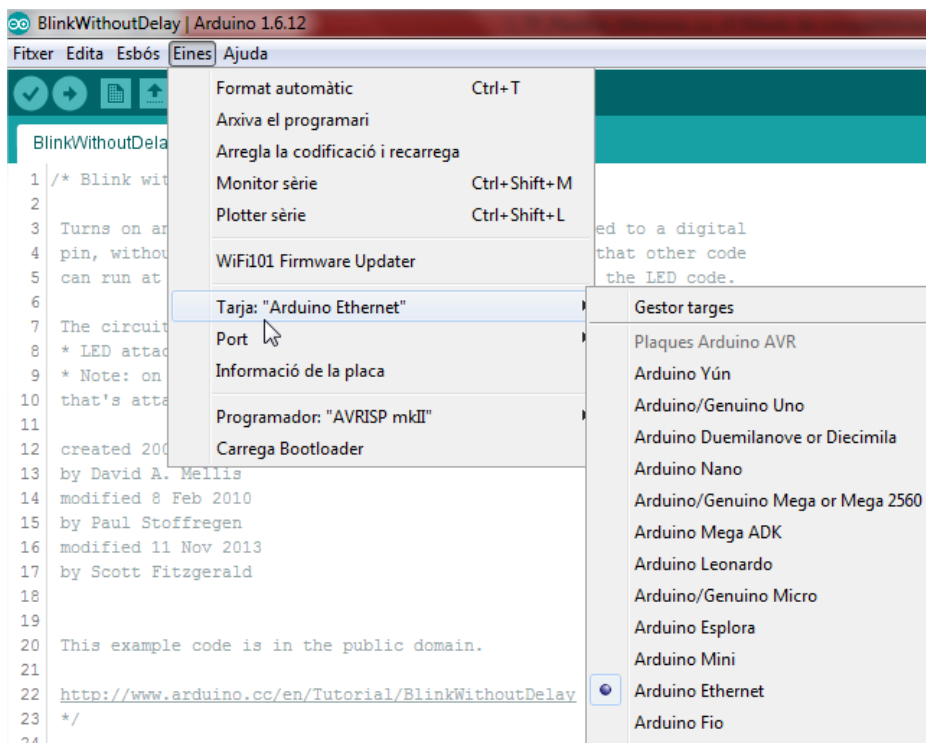


Figura 14 selecció de model d'Arduino en l'IDE d' Arduino

2.3.2 Llenguatge de programació

Arduino utilitza un llenguatge propi basat en el llenguatge de programació d'alt nivell anomenat Processing ,que es basa en C++

L'estructura del programa

Com a mínim cada programa disposa de 2 parts :

- Funció de configuració
- Funció de bucle.

En la funció de configuració o `setup()` , se situa en la part inicial del programa i s'utilitza per inicialitzar els modes de treball dels pins (entrada o sortida), el port sèrie, i paràmetres de xarxa com l'adreça `IP` , màscara de `subxarxa` i porta d'enllaç en cas que es treballi amb algun mòdul `Ethernet`.

La funció de bucle o `loop()`, fa la crida a un bucle infinit que contindrà l'execució continua dels esdeveniments que s'han de produir en la placa.

2.3.3 Llibreries

El sistema de programació de la plataforma Arduino, permet afegir mitjançant llibreries funcionalitats extres pel que fa a maquinari i manipulació de la informació. Per defecte el IDE incorpora de sèrie múltiples llibreries com les que permeten la programació amb plaques Ethernet, treballar amb teclat o ratolí,etc.

A part múltiples fabricants de maquinari per Arduino (sensors, Shields, motors, etc.) aporten llibreries que es poden importar que permeten treballar amb els seus productes. A més per internet es poden trobar llibreries desenvolupades per altres usuaris.

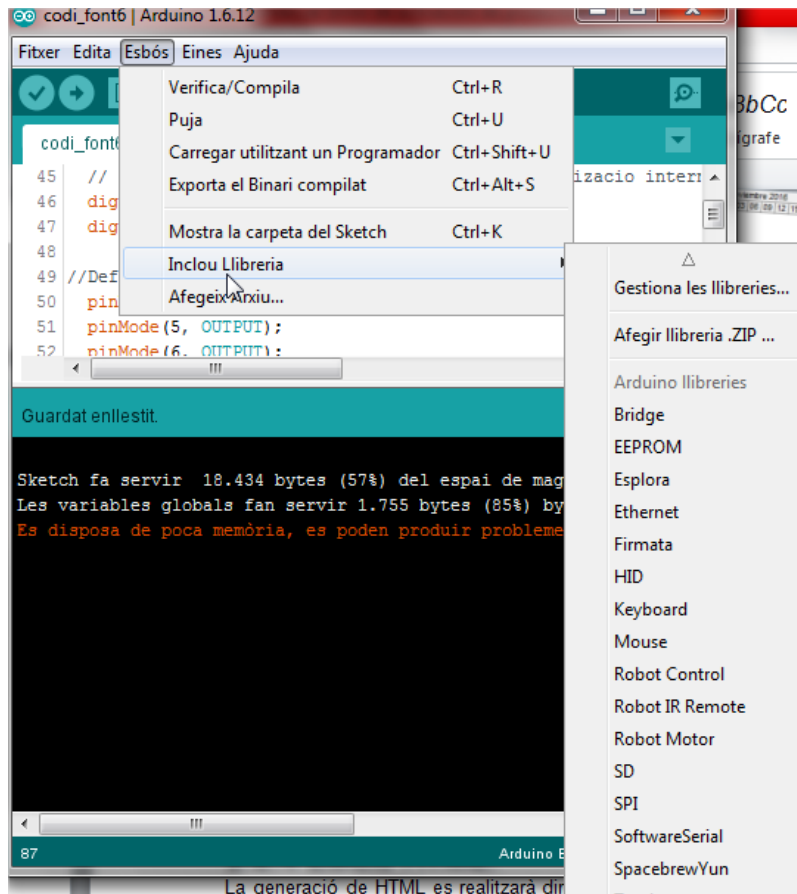


Figura 15 selecció de llibreries per carregar a l'Sketch

2.3.4 Funcions bàsiques i operadors

Pel fet que el llenguatge de programació de l'IDE d'Arduino està basat en C l'estructura de la sintaxi i funcions són molt similars a C.

Entre els diferents elements que formen la sintaxi tenim:

- Delimitadors: com el ; o {}
- Obrir comentaris amb // per una sola línia o be /* i */ per obrir un bloc de comentaris
- Capçaleres com #define
- Assignació =
- Operadors + - / * (aritmètics) == < > => (comparadors) += increments

El llenguatge d' Arduino disposa d'estructures de control com:

- Condicionals : *if* , *if..else*, *switch case*
- Bucles : *for*, *while*, *do..while*

Es disposa de variables, constants i tipus de dades (*boolean*, *int*, *char*, *string...*)

Per E/S digital es pot establir el control de les entrades amb les comandes:

- *pinMode(pin, mode)*: Permet especificar el mode (entrada o sortida) d'un pin digital
- *digitalWrite(pin, valor)*: Escriu el valor (LOW o HIGH) a un pin en concret
- *int digitalRead(pin)*: Llegeix el valor d'un Pin en concret

I per l'E/S analògica digital hi ha les comandes:

- *analogReference(tipus)*:Especifica el voltatge de referència amb què es treballarà el pin d'entrada
- *int analogRead(pin)*: Llegim el valor analògic d'un pin

Pel control de temps es disposa de:

- *unsigned long millis()*: ens retorna el número de mil·lisegons que porta l'Arduino des que ha començat a treballar amb el programa actual
- *unsigned long micro()* : Ens retorna el número de microsomes des que es va iniciar l'Arduino amb el programa en curs.
- *delay(ms)* : fa una pausa dels mil·lisegons assignats.
- *delayMicroseconds(microsegons)* : Fa una pausa del nombre de microsegons especificats.

2.3.5 Edició HTML

La generació de HTML es realitzarà dinàmicament en el bucle d'execució del programa d'Arduino, tanmateix i per facilitar la tasca, es dissenyaran estèticament les pàgines que caldrà emprar. Una vegada fet s'utilitzarà el codi HTML de la pàgina per generar-se dinàmicament.

S'utilitzarà un programa senzill i gratuït per permetre l'elaboració de pàgines simples.

S'ha optat per l'aplicació KompoZer, per la seva senzillesa i per ser un editor WYSIWYG (What You See Is What You Get).

2.4 Disseny del projecte:

2.4.1 A nivell de Hardware:

Segons les necessitats del projecte s'ha determinat els següents requisits a nivells de ports digitals i analògics.

Ports de Sortida:

Ens farà falta 4 ports de sortida per controlar els següents valors:

- Port Porta de Vianants: Aquest port controlarà un relé que farà tancar un circuit elèctric que activarà alhora un altre relé que obrirà la porta de vianants, el relé tanca el circuit durant 1 segon.
- Port Porta de Vehicles: Aquest port controlarà un relé que tancarà un circuit elèctric que activarà alhora un altre relé que obrirà la porta de Vehicles. El relé tanca el circuit durant 1 segon.
- Ordinador: Ens permetrà controlar un relé que activarà un relé que tancarà el circuit que connecta amb el "jumper" d'interruptor d'un ordinador. Tindrà 2 funcionalitats : Obrir i Tancar l'ordinador. En cas que es vulgui engegar l'ordinador, el relé tancarà el circuit durant un segon i en cas que es vulgui apagar, aquest tancarà el circuit durant 5 segons.
- Ventilació: Aquest port controla un relé que tanca el circuit d'un ventilador, des de l'Arduino es podrà tancar o obrir segons ens convingui.

Ports d'entrada digitals:

En els ports d'entrada digital detectarem diferents variacions de voltatge entre $-V_{cc}$ i $+V_{cc}$ sense passar per l'interval de valors que hi ha entre aquest voltatge. Per tant disposem de dos valors de senyals que ens indiquen dos estats. Al nivell de tensió $-V_{cc}$ Arduino li associa el valor lògic LOW o 0 i en l'altre voltatge se li associa el valor HIGH o 1.

En l'àmbit físic no es disposa de voltatges discrets on es passi del valor inferior al superior, si no que hi ha un increment gradual de voltatge. De manera que Arduino i altres dispositius amb entrades digitals determinen cert voltatge llindar que marcarà la diferència entre el que serà $-V_{cc}$ i $+V_{cc}$. Per poder disposar d'un sistema de 2 valors.

En Arduino els voltatges de treball són 0V i 5V de manera que si es detecta un voltatge entrant proper a 5V el valor de la lectura serà el valor HIGH, en canvi si el voltatge d'entrada és proper a 0 V la lectura obtinguda serà LOW. En Arduino el voltatge llindar sol situar-se a 2,5 V i si la lectura és propera a aquest valor es podria obtenir una lectura incorrecta, ja que si hi ha variacions lleugeres de voltatge podrien induir lectures errònies.

En el prototip simularem les entrades digitals com la interrupció o connexió d'un circuit de 5 V situant un interruptor enmig del circuit.

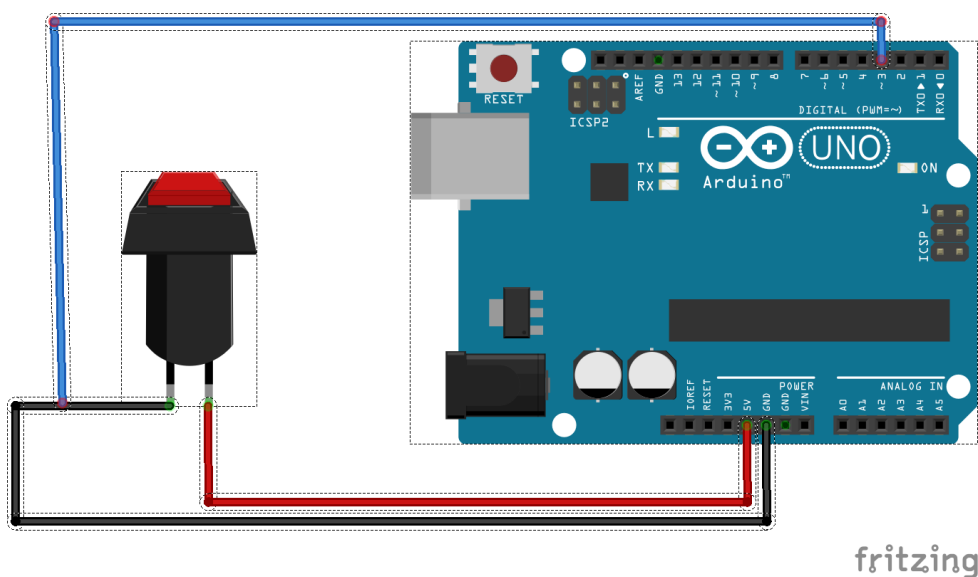


Figura 16 Esquema basic de connexió d'entrada digital amb interruptor.

Ara bé, cal introduir en el circuit una resistència a fi d'evitar possibles curtcircuits i depenent de la ubicació de la resistència amb la posició de l'interruptor podem fer que l'entrada digital del circuit marqui el valor HIGH o LOW quan es tanqui per mitjà de l'interruptor. Segons la posició de la

resistència amb el l'interruptor i l'interruptor es tindrà una resistència PULL DOWN o una resistència PULL UP.

Aquestes resistències es connecten entre l'entrada digital i una de les tensions de referència (5v o 0V) de manera que es força que el valor de la tensió segons s'utilitzi la resistència PULL UP o PULL DOWN

La resistència PULL UP té situat la resistència entre el punt de voltatge elevat i l'entrada digital, i l'interruptor se situa posteriorment, més a prop de terra. D'aquesta manera quan l'interruptor està obert el corrent es dirigeix del punt de voltatge més elevat cap a l'entrada digital, de manera que la lectura obtinguda serà HIGH. Quan l'interruptor estigui tancat, el corrent tendirà a anar de l'entrada digital cap a terra de manera que el valor serà LOW.

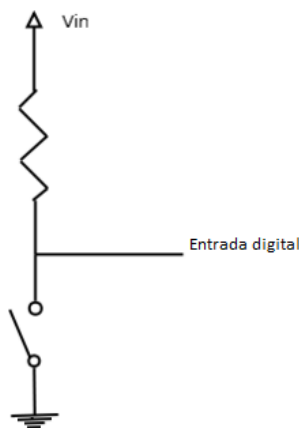


Figura 17 Esquema de circuit d'entrada digital amb resistència PULL UP

La Resistència PULL DOWN és el que té la resistència situat entre l'entrada digital i el punt de terra i l'interruptor està entre el pin d'entrada i la tensió més elevada. I per la facilitat de circulació dels electrons quan el circuit estigui obert el valor serà LOW i quan estigui tancat tindrà el valor de HIGH.

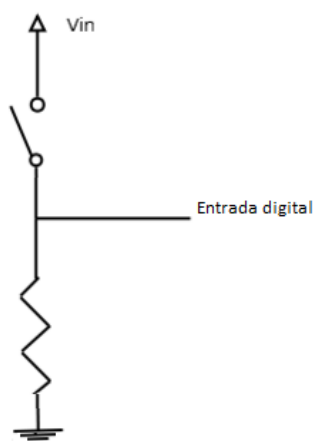


Figura 18 Esquema de circuit d'entrada digital amb resistència PULL DOWN

En els casos que en què treballem interessarà que quan es tanqui el circuit li entri el valor HIGH de manera que treballarem amb resistència de tipus PULL DOWN.

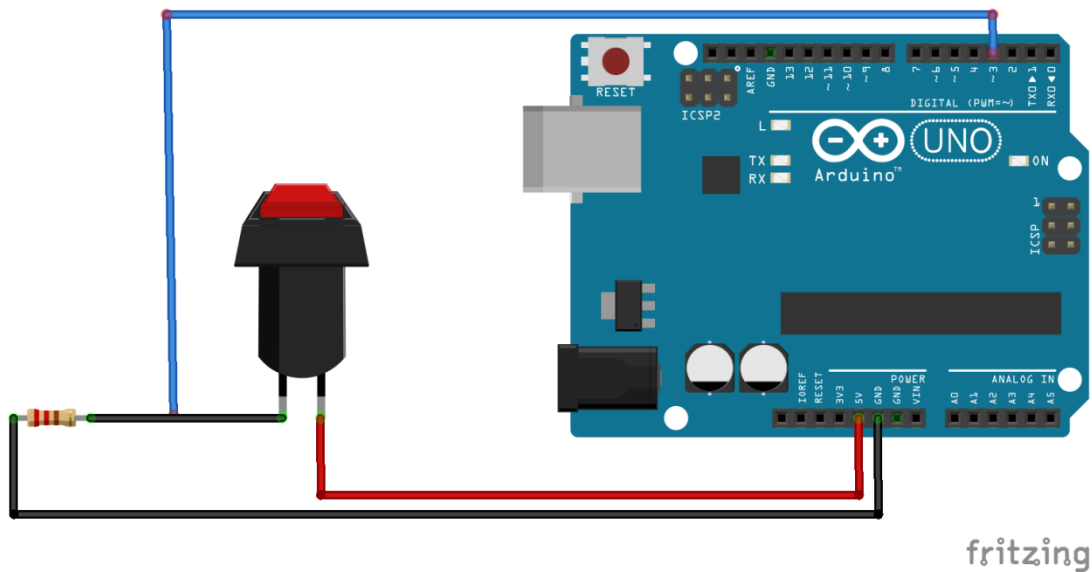


Figura 20 Esquema basic de connexió amb resistència PULL UP.

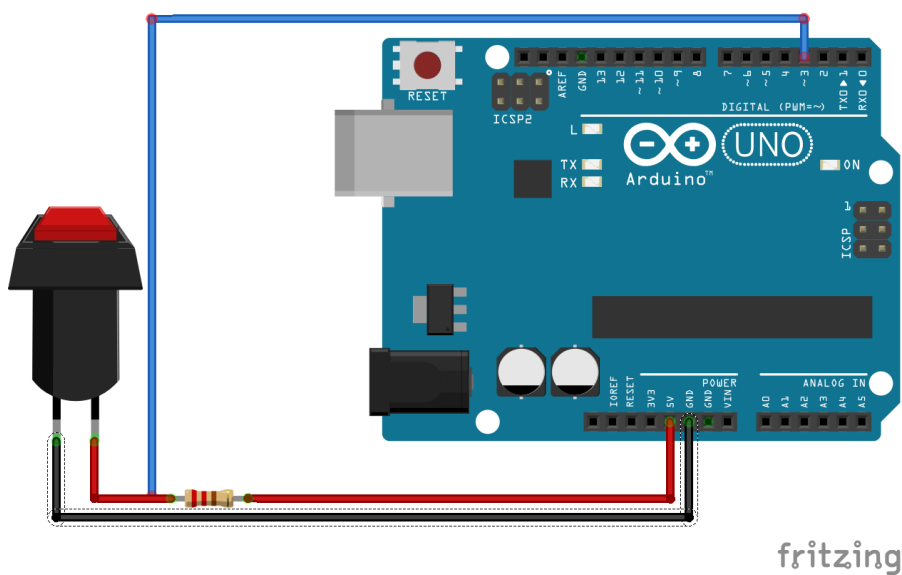


Figura 21 Esquema basic de connexió amb resistència PULL DOWN.

- Alarma de foc: Quan es detecta un foc (per mitjà dels sensors) o bé algú ha trencat el vidre de l'interruptor per advertir de foc, hi ha un relé que tanca un circuit elèctric, que dispara el timbre de l'alarma. Si es connecta el cable per la part carregada a un port digital detectarà si hi ha carrega,

en aquest cas el sistema Arduino detectarà que l'alarma de foc està activada.

En la ubicació final es podrà provar forçant l'alarma o bé prement l'interruptor (sense trencar el vidre, només traient el cargol i apartant el vidre).



Figura 22 Interruptor per prémer en cas de detecció de focs

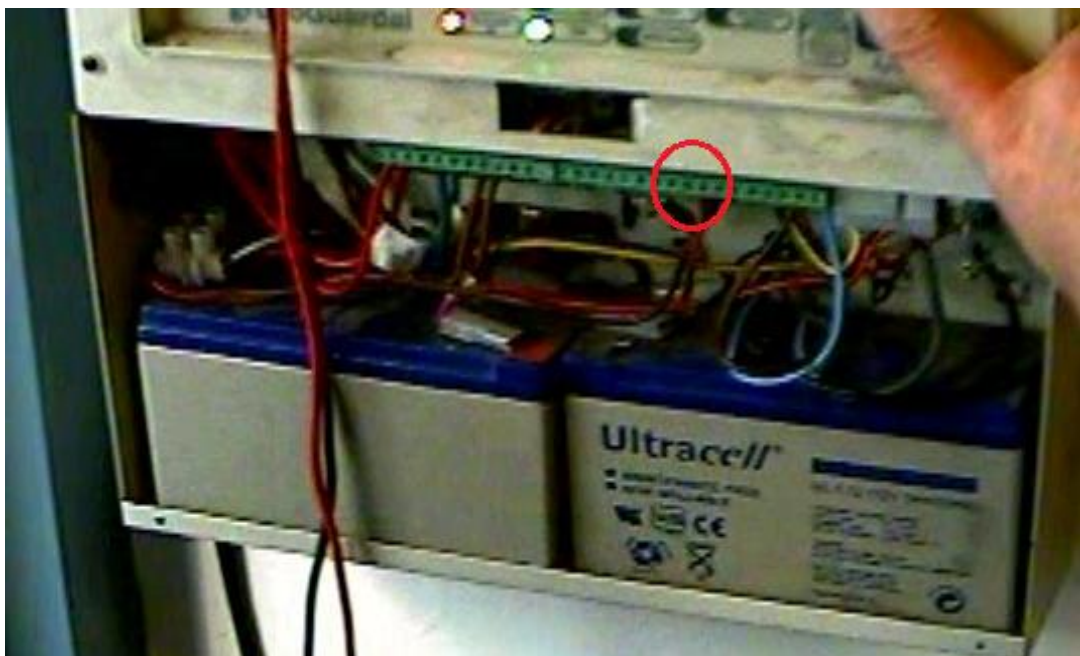


Figura 23 Part no visible de l'alarma d'incendis on hi ha les connexions on es connectaran amb Arduino en el pin de FOC

- Control de pressió de l'aigua: El sistema és molt bàsic, hi ha un pressòstat que està calibrat a una pressió concreta i en cas que hi hagi una pèrdua de pressió, hi ha un desplaçament en l'aparell i es tanca un circuit elèctric. El port d'aigua detectarà que hi ha carrega i indicarà que hi ha una pèrdua de pressió.

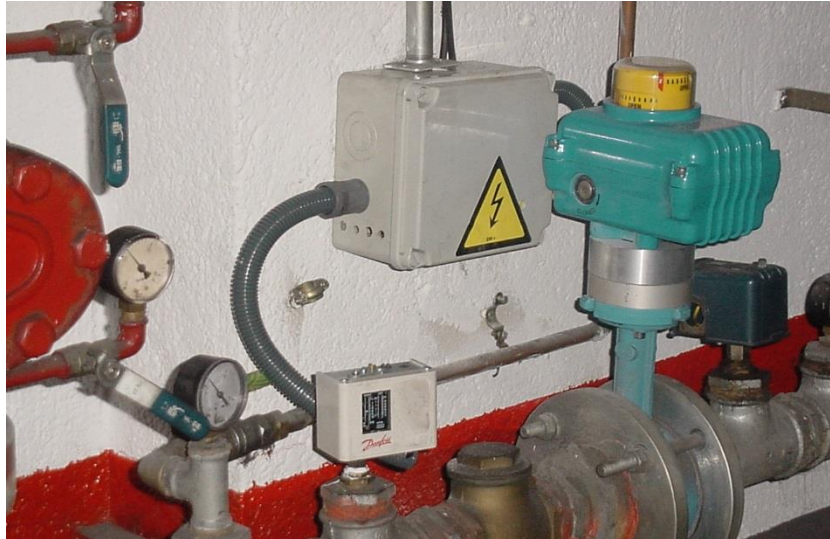


Figura 24 Canalització d'aigua i el pressòstat on es connecta amb els pins d'aigua

Ports d'entrada Analògics:

- Sensor de Monòxid de Carboni: El sensor escollit que s'emprarà és el sensor MQ-7. Es realitzarà una descripció més completa d'aquest dispositiu en el proper apartat d'aquest capítol..



Figura 25 Sensor model v.1.3

2.4.1.1 Sensor Analògic de Gasos MQ7

Aquest sensor és membre d'una família de sensors desenvolupats per Sparkfun Electronics <https://www.sparkfun.com> ,Cada un dels sensors de la família MQ té sensibilitat a diferents gasos, MQ2 té sensibilitat al butà, Metà i fum i el sensor MQ3 a Etanol, Metanol .



Figura 26 Sensor MQ2 d'Sparkfun

En aquest projecte s'utilitza un sensor MQ7 connectat ja a una placa que facilita la connexió a l'Arduino. Aquest circuit té un connector per alimentar el circuit a 5 V, un connector per la connexió de terra, un d'analògic per l'entrada analògica i un de digital.

En el projecte actual s'emprarà el connector analògic. Per a poder determinar el valor en PPM (parts per milió) en la concentració de CO amb el sensor MQ-7 en primer lloc s'examina el "DataSheet" del sensor MQ-7 per veure quina relació hi ha entre el valor obtingut en la lectura del sensor analògic respecte la concentració.

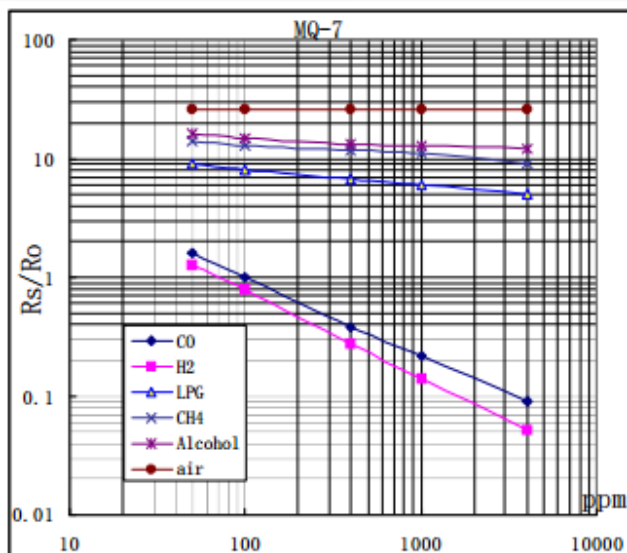


Fig.3 is shows the typical sensitivity characteristics of the MQ-7 for several gases.

in their: Temp: 20°C,

Humidity: 65%,

O₂ concentration 21%

RL=10k Ω

Ro: sensor resistance at 100ppm CO in the clean air.

Rs: sensor resistance at various concentrations of gases.

Figura 27 Gràfic on indica la relació del PPM de diferents gasos amb RS/R0 pel model de sensor MQ7

A partir d'aquí es pot fer una taula dels valors relacionant la concentració entre RS/Ro

PPM	rs / ro
50	1,7
100	1
200	0,6
400	0,4
1000	0,21
4000	0,09

Taula 2 Relació PPM amb RS/RO extreta de la figura16

Amb aquesta taula es pot fer la gràfica i la recta de la línia de tendència. I la seva equació de la recta.

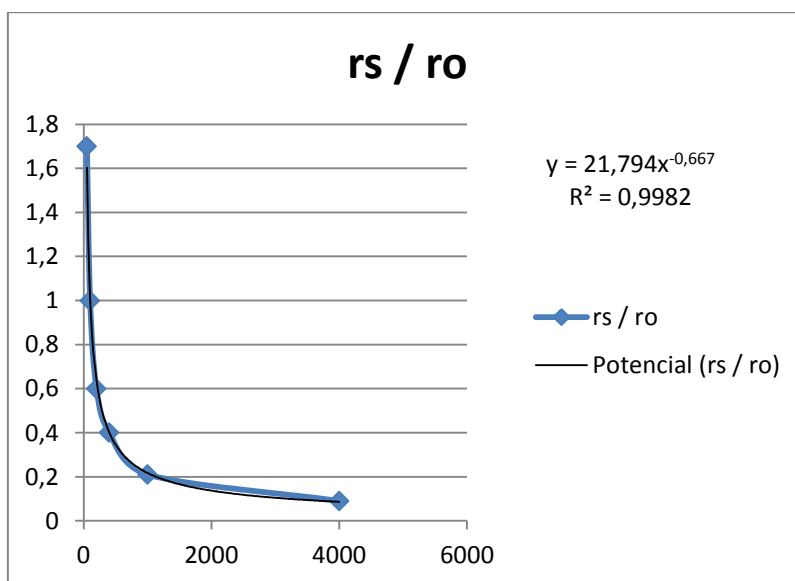


Figura 28 Línia de tendència a partir dels punts de la taula 2

A fi de simplificar la recta, ja que l'equació i, ja que es treballa a una escala logarítmica. Es torna a calcular amb el logaritme:

PPM	rs / ro	ln ppm	ln rs/ro
50	1,7	3,91202301	0,53062825
100	1	4,60517019	0
200	0,6	5,29831737	-0,51082562
400	0,4	5,99146455	-0,91629073
1000	0,21	6,90775528	-1,56064775
4000	0,09	8,29404964	-2,40794561

Taula 3 Modificació de la taula 2 obtenint els valors de PPM i RS/RO en escala logarítmica per simplificar l'equació exponencial

La gràfica resultant dels nous valors a escala logarítmica i l'equació de la recta de tendència seran:

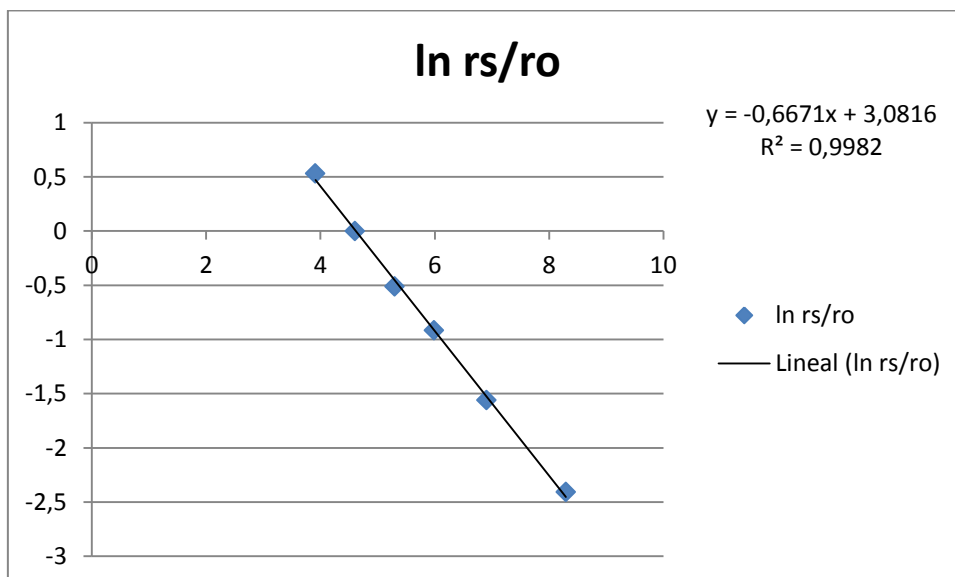


Figura 29 Línia de tendència a partir dels punts de la taula 3

Es pot obtenir els valors puntuals de rs segons el voltatge que tingui la lectura analògica, si tenim en compte que el sensor funciona com un divisor de corrent on la corrent de sortida depèn de la corrent d'entrada (5V) i de 2 resistències . Una de les resistències és la que depèn de la concentració de gasos i l'altra és fixa.

El voltatge de sortida es podrà obtenir a partir d'aquesta equació.

$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

Figura 30 Equació del divisor de voltatge

Segons el "datasheet "de MQ-7 la resistència R_1 és de $10K\Omega$.

A falta d'un laboratori o un sensor de CO homologat per calibrar el MQ7 a partir de l'obtenció del valor fix de R_0 es faria el calibratge en l'aire lliure.

Es coneix pel "datasheet" que $R_0=R_S$ en una concentració de 100 PPM de CO però s'ignora quina és la concentració en el moment de fer la mediació de manera

Una vegada s'hagi conegut els valors de PPM ambientals es pot determinar quina és la lectura, mitjançant la lectura amb el següent codi:

```

1 void setup() {
2   Serial.begin(9600);
3 }
4 void loop() {
5
6   float voltatge=0;
7   int adc_MQ7 = analogRead(A0); //Entrada analogica de MQ7
8   voltatge = adc_MQ7 * (5.0 / 1023.0); //Convertim el valor de la lectura a en voltatge obtingut
9   delay(100);
10  Serial.print(" lectura analogica:");
11  Serial.print(adc_MQ7);
12  Serial.print("   voltatge:");
13  Serial.print(voltatge);
14  float Rs=10000*((5-voltatge)/voltatge); //calculem la resistencia generada per el detector assumint que RL=10k Ohm
15  Serial.print("   rs:");
16  Serial.println(Rs);
17
18  delay(100);
19 }

```

Figura 31 Codi per obtenir el valor de RS suposant que la resistència de carrega és de 10K Ohm

El valor de V_{out} s'obte a partir del valor obtingut de la lectura analògica multiplicat pel voltatge dividit per 1023 (ja que la lectura és un int i el rang de valors que pot tenir la lectura van de 0 a 1023)

I el voltatge determinat pel sensor analògic ve donat per un divisor de corrent format per 2 resistències, una de càrrega i la que determina la concentració de CO.

2.4.1.2 Relés

Un relé és un dispositiu electromecànic que funciona com un interruptor controlat per un circuit elèctric. D'aquesta manera es pot connectar o desconnectar un circuit aliè al circuit on hi ha connectat el relé.

Pel que fa a aquesta pràctica, els relés emprats serviran per actuar com interruptors que activa l'obertura de la porta de vehicles i la porta de vianants, per tancar el circuit que fa d'interruptor a la placa base de l'ordinador de càmeres i per encendre i apagar la ventilació.

2.4.1.3 Estructura general del circuit

Una vegada s'han connectat tots els connectors en els pins d'entrada digitals per la detecció de l'alarma de foc i el canvi de pressió. L'entrada del sensor analògic i la sortida digitals per activar- desactivar els relés L'estructura del circuit serà similar a la següent.

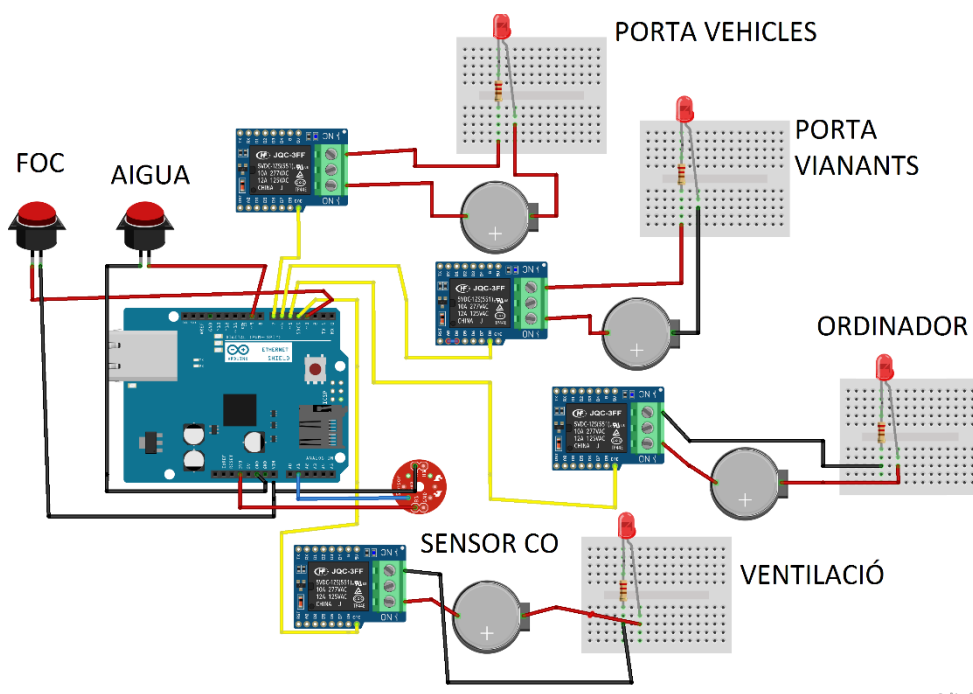


Figura 32 Diagrama fet amb Fritzing de les connexions de l'Arduino amb les entrades digital , sortides digitals per relés i sensor analògic.

Cal tenir en compte que tan el senyal de foc com d' aigua , una vegada s' activi el senyal es tancarà un circuit que farà passar un voltatge de 5V que entrarà a la seu respectiu pin d' entrada.

Pel que a les sortides, en la imatge apareix com a relés aïllats, de fet la connexió real serà per mitja d' un Shield de relé , de 4 relés i del que per mitja dels pin digitals 4,5,6 i 7 es podrà activar un dels diferents dispositius, cada un d' ells estarà connectat o bé a la porta de vianants , a la porta de vehicles , l' ordinador o a la ventilació.

2.4.2 A nivell pàgina Web:

El Sistema Arduino en mostrarà l'estat dels pins de sortida en una pàgina web que es generarà a cada cicle del bucle.

Quan estigui tot correcte el que ens mostrarà tindrà la següent estructura:

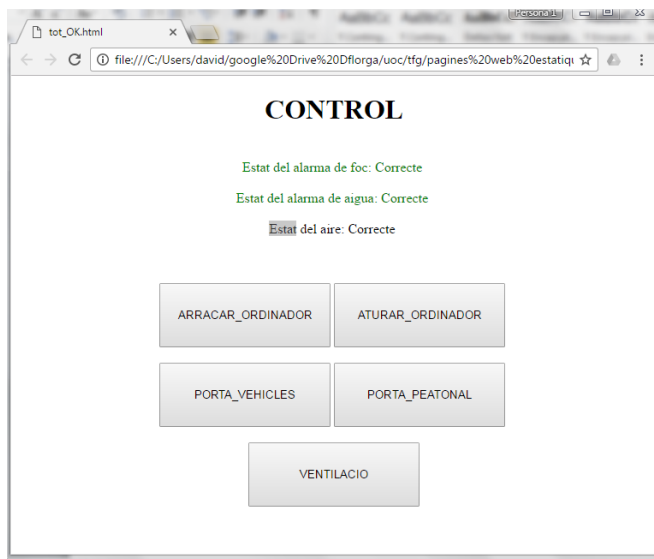


Figura 33 Captura de la pàgina Web generada per Arduino en el prototip inicial on tot és correcte

En cas que s' activi una o varies alarmes els textos canviaran de contingut i de color, en verd per indicar estat correcte i en vermell quan s' ha superat el llindar establert.

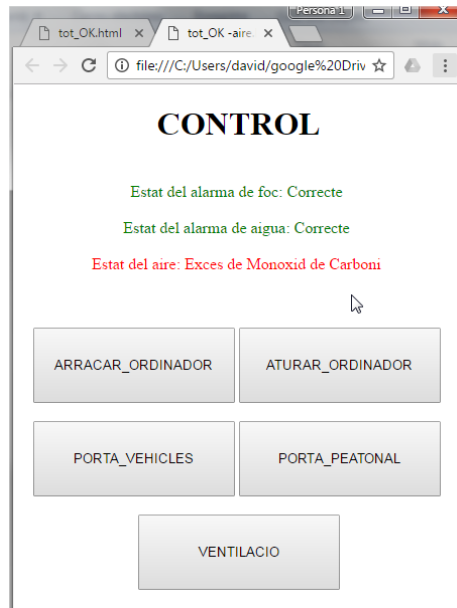


Figura 34 Captura de la pàgina Web generada per Arduino en el prototip inicial on hi ha excés de Monòxid de Carboni.



Figura 35 Captura de la pàgina Web generada per Arduino en el prototip inicial on hi ha excés de Monòxid de Carboni i un canvi de pressió d'aigua..

En cas de prémer un dels botons s'activarà els relés que estan connectats als pins de 4 al 7 de l' Arduino amb les següents funcions:

- Arrancar Ordinador: Engegar el relé connectat el pin 4 durant un segon.
- Aturar Ordinador: Engegar el relé connectat el pin 4 durant 5 segons.
- Porta vehicles: Engegar el relé connectat el pin 5 durant un segon.
- Porta_Peatonal: Engegar el relé connectat el pin 6 durant un segon.

- Ventilació: Canviar l'estat del relé connectat el pin 7

Posteriorment a partir de l'entrega de la PAC2 s'ha realitzat una millora en la pàgina web.

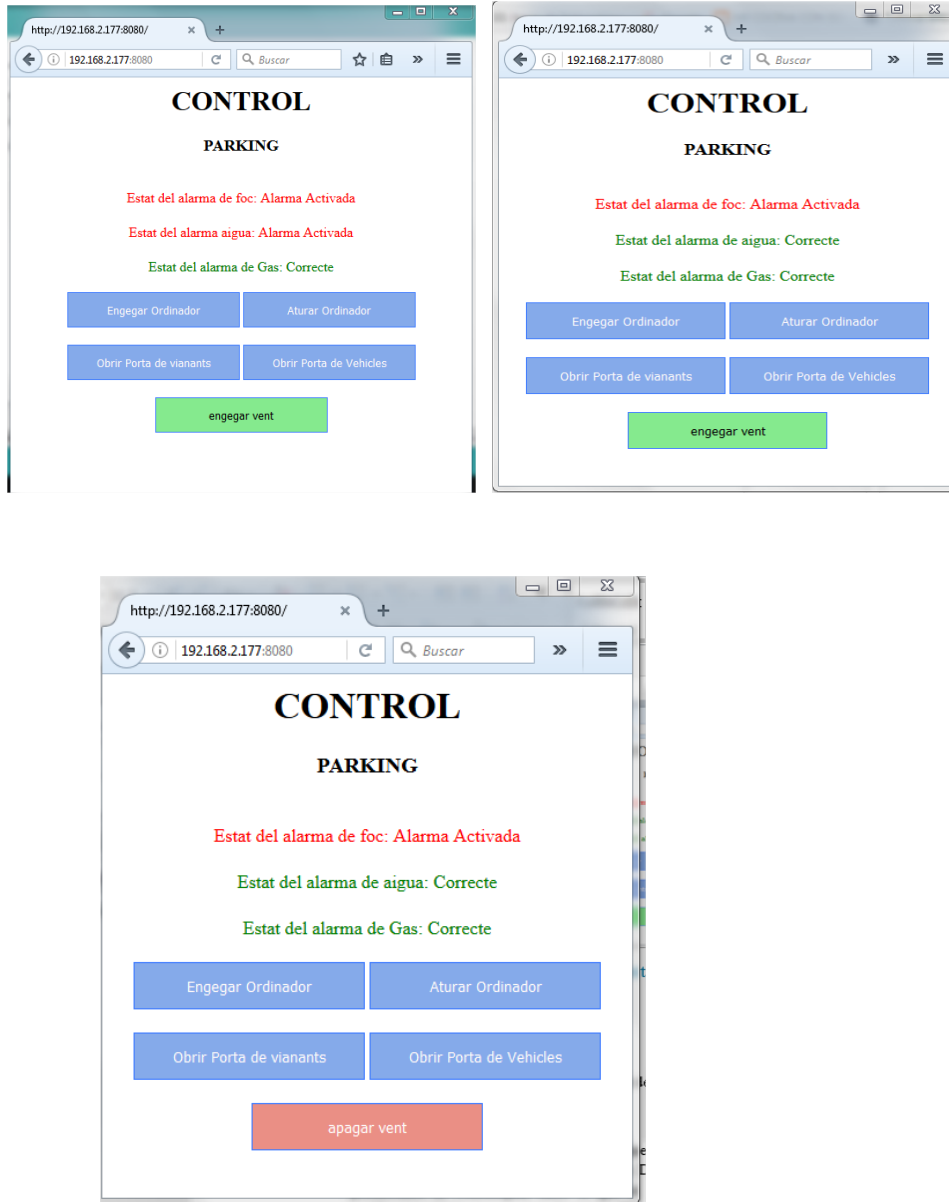


Figura 36,37,38 Captura de la pàgina Web definitiva generada per Arduino amb diferents estats

2.4.3 Mecanismes de notificació.

Es disposen diferents mètodes d'enviament de notificacions per a l'usuari com podria ser per mitjà de SMS, notificació per Whatsapp, entrades a Twitter o enviament de correu electrònic.

En aquest apartat es valorarà les diferents opcions i s'explicarà quin és el que finalment es tria.

- SMS: Servei de Missatges curts de la telefonia mòbil. Té com a punts a favor la immediatesa i la simplicitat i no cal que el destinatari disposi d'una connexió a internet. Com a punts en contra: Cal disposar de maquinari extra per establir la connexió entre Arduino i la targeta GSM i pagar un cost per cada missatge o contractar un paquet de missatges.
- Whatsapp. Sistema de missatgeria per internet. Similar a SMS però cal disposar de connexió a internet. No cal disposar de maquinari extra però cal contractar un paquet extra per poder enviar missatges des del nostre dispositiu.
- Twitter: Es disposa de llibreries que permeten enviar notificacions d'Arduino a Twitter però cal que l'usuari estigui connectat a l'aplicació per veure si hi ha notificacions.
- Correu Electronic: No cal disposar d'un maquinari extra a part que s'ha adquirit i depenent del sistema caldrà contractar un servei o bé disposar d'un servidor SMTP disponible per a poder contactar i enviar els correus. Com a punts febles cal que l'usuari estigui connectat a internet per a comprovar que hi ha notificacions.

Finalment i després de valorar les diferents opcions s'ha escollit com a sistema de notificacions el correu electrònic, ja que és l'opció més econòmica perquè no cal adquirir nou material, ni subscripcions per enviar correu, ja que es disposa d'un servidor SMTP.

Hi ha diverses formes d'enviar un correu, depenent del servidor que es vulgui emprar ens caldrà utilitzar una d'elles:

- Temboo: Tembo és una plataforma de pagament que serveix com intermediari per connectar-se amb diferents aplicacions web. De manera que permet que diferents plataformes es comuniquin entre si que et permet realitzar diferents tipus de notificacions ja sigui per correu electrònic, SMS. Ja que inclou llibreries que fan que des d'Arduino es pugui utilitzar les Apis de Twitter, Facebook, Instagram, MySQL, Gmail, etc. Cal tenir un compte creat i segons el tipus de notificació et genera el codi per efectuar l'enviament, que després podrà ser copiat i enganxat a l'Sketch.
Cal descarregar-se la llibreria i hi ha un nombre limitat d'enviament segons què es tingui contractat.

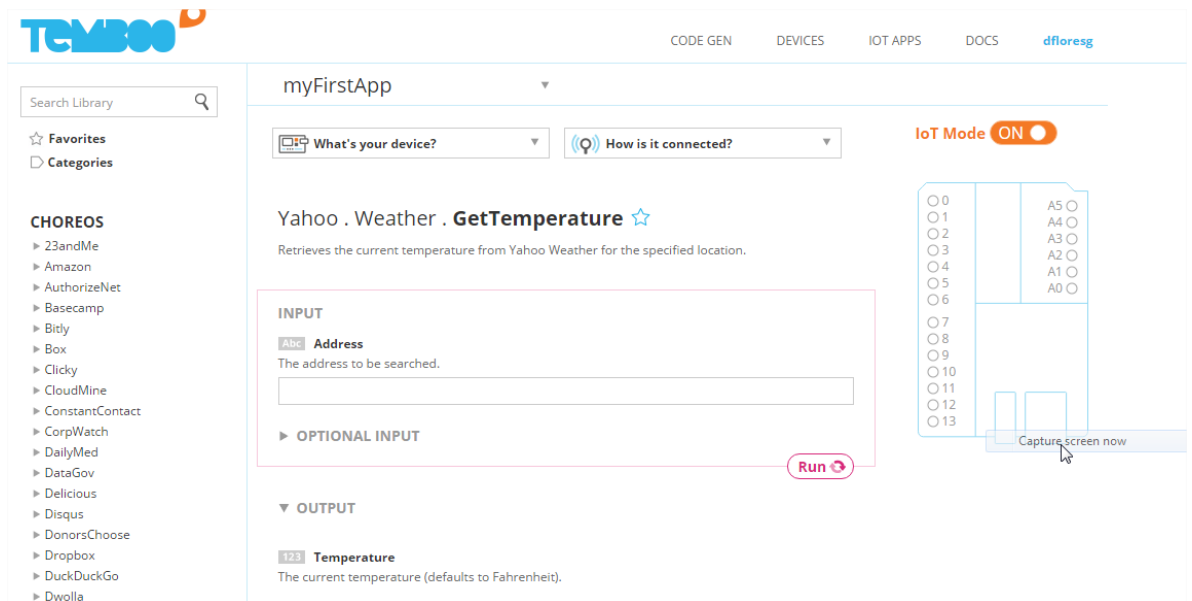


Figura 39 Captura de pantalla de la pàgina Web de Temboo.

- Mitjançant una sessió SMTP. Es pot realitzar una connexió TCP on s'utilitza el port 25 (per defecte) amb un servidor SMTP, on es validarà amb un usuari i paraula de pas, després s'emprarà les comandes per establir la sessió SMTP, aquesta consta dels següents passos:
Establir una connexió SMTP amb el servidor

Telnet <nom_servidor o IP> 25

EHLO *(identificador d'equip i domini) aquesta comanda serveix per identificar al servidor.*

AUTH LOGIN *(amb aquesta comanda s'indica que es vol una sessió autenticada amb usuari i "password") , on el servidor respondrà amb una petició xifrada de l'usuari*

NOM_USUARI *codificat en base 64*

PASSWORD *codificat en base 64.*

MAIL FROM: <remitent@domini> *s'indica que es desitja enviar el correu des de l'adreça especificada en el camp del remitent.*

RCPT TO: <destinatari@domini> *en aquesta comanda s'especifica a qui s'envia el correu.*

DATA :Amb aquesta comanda indica que els que hi ha a continuació és el contingut de text a enviar i es finalitzarà el contingut amb un salt de línia, un punt (.) i un salt de línia.

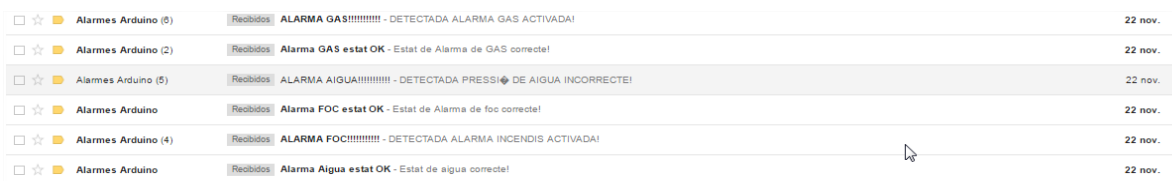
QUIT : Amb aquesta comanda es tanca la sessió SMTP.

En aquest cas inicialment es volia emprar un sistema intermediari com Temboo, ja que es volia utilitzar un compte de correu electrònic de GMAIL, però ja que s'ha pogut aconseguir un compte d'un servidor SMTP s'ha preferit utilitzar un sistema més estàndard d'enviament de correu i no és necessari mantenir un compte a una aplicació externa que podria deixar de funcionar en qualsevol moment.

En aquest cas des d'Arduino, es crea una connexió TCP amb el servidor de correu en el port 25. A continuació s'exposa unes línies de codi utilitzar per enviar comandes de SMTP a través de l'Arduino:

```
if(client.connect(server,25)) {  
  Serial.println(F("connected"));  
} else {  
  Serial.println(F("connection failed"));  
  return 0;  
}
```

Com s'ha vist, s'anirà introduint les comandes de la sessió de SMTP mitjançant transferint-les mitjançant el mètode println de la connexió que s' ha creat amb el servidor de correu amb l'ordre client.println



<input type="checkbox"/>	☆	Alarmes Arduino (6)	Rebidos	ALARMA GAS!!!!!!!!!!!! - DETECTADA ALARMA GAS ACTIVADA!	22 nov.
<input type="checkbox"/>	☆	Alarmes Arduino (2)	Rebidos	Alarma GAS estat OK - Estat de Alarma de GAS correcte!	22 nov.
<input type="checkbox"/>	☆	Alarmes Arduino (5)	Rebidos	ALARMA AIGUA!!!!!!!!!!!! - DETECTADA PRESSIÓ DE AIGUA INCORRECTE!	22 nov.
<input type="checkbox"/>	☆	Alarmes Arduino	Rebidos	Alarma FOC estat OK - Estat de Alarma de foc correcte!	22 nov.
<input type="checkbox"/>	☆	Alarmes Arduino (4)	Rebidos	ALARMA FOC!!!!!!!!!!!! - DETECTADA ALARMA INCENDIS ACTIVADA!	22 nov.
<input type="checkbox"/>	☆	Alarmes Arduino	Rebidos	Alarma Aigua estat OK - Estat de aigua correcte!	22 nov.

Figura 40 Captura de pantalla on apareixen els correus rebuts com a notificacions.

2.5 Problemes detectats:

Entre els diferents problemes que s'han detectat en el desenvolupament del projecte es farà incís dels següents:

1. Component cremat per possible curtcircuit: Al connectar el sistema de relés connectat a una sèrie de LEDS i emprar un transformador de corrent directe de 12 V per alimentar la placa Arduino, un component va començar a cremar-se apreciand-se l'olor de component malmès i fum. A més a partir d'aquell moment quan es connectava la placa amb el transformador alguns LEDs funcionaven però no es rebia senyal a través de Ethernet o port sèrie . Tampoc funcionava si la corrent elèctrica venia pel port FTDI . Realitzant diverses proves es va detectar que si es proporcionava corrent a través de l'alimentador directe i pel FTDI els diferents LEDs de control de la placa funcionaven i el programa corria perfectament. Responien els ports de sortida i els ports d'entrada donaven resposta , a més es tenia resposta pel port sèrie i Ethernet.

Pel que es pot veure, aquesta placa només pot funcionar si s'alimenta pel punt d'alimentació directa i pel connector FTDI.

2. Durant el desenvolupament de les diferents parts del programa no s'han detectat problemes seriosos a part dels típics errors de sintaxi que fàcilment eren corregits. Però en integrar totes les parts en una de sola s'ha detectat que en afegir el mòdul de lectura analògica deixaven de funcionar els pins de sortida, quan es premia el botó d'execució en la pàgina web cap dels relés feia el que havia de fer. Aquest problema també s'ha detectat en voler fer la pàgina web més elaborada (afegir fons de color o fer que el botó de ventilació canviés el títol segons el valor d'una variable). Finalment s'ha deixat la pàgina funcionant però sense poder deixar-se com es voldria a fi prioritzar l'execució de la sortida digital.

Aquesta incidència ha reportat que s'hagi invertit més temps de l'esperat per assolir aquesta fase com que no es trobava la raó per la qual quan s'augmentava la complexitat de la pàgina Web deixava de funcionar el sistema de botons. Això pot ser degut a diverses causes:

En compilar ens informa que l'espai de memòria emprat per les variables arriba entre el 85 i 90% de la capacitat màxima de memòria disponible.

- Una altra causa podria ser conseqüència del curtcircuit esmentat en l'apartat anterior. Però sense una altra placa per contrastar-ho no es pot demostrar.

Finalment, entre l'entrega de la PAC2 i la PAC3 es va poder reduir l'espai demandat per les variables en la memòria dinàmica modificant les cadenes de text que generen la pàgina web, fent que aquesta informació quedi desada en la memòria del programa en lloc de la memòria RAM. Per a fer-ho es va modificar les cadenes com la següent:

```
WebClient.println("<!DOCTYPE html>"); Per  
WebClient.println(F("<!DOCTYPE html>"));
```

D'aquesta manera s'ha pogut millorar notablement l'estètica de la pàgina Web sense perdre funcionalitats amb l'ús dels botons..

3. S'ha volgut crear un compte de correu electrònic de Gmail expressament per a realitzar l'enviament del correu electrònic per avisar del canvi d'estat de les diferents entrades digitals, Però a causa de la complexitat amb què es treballa no he pogut efectuar l'enviament de correu amb el servidor de correu electrònic de Gmail mitjançant una sessió de Telnet per mitjà de SMTP. Per tant després de diferents intents s'ha creat un compte de correu a través d'un ISP que gestiona el domini de l'empresa on s'ha d'implementar finalment el projecte. Aquest ISP permet fer un enviament de correu a través de comandes SMTP autènticat.
4. De nou un curtcircuit produït el 18 de desembre ha fet que es malmetin els ports de sortida digitals 8 i 9 i que el port 3 només funcioni amb la polaritat baixa (es posa en estat HIGH quan no té corrent). s'ha optat per adquirir una nova placa per a poder realitzar la depuració del programari correctament.

2.6. Valoració econòmica

Pel que fa al material necessari per a la realització del prototip hi ha el desgloss següent:

Component	Cost de material
Placa Arduino Ethernet	42,90 €
Arduino relay Shield	17,60 €
Arduino Screwshield	12,40 €
Sensor MQ7	9,95 €
Cables de configuració i verificació	20,00 €

Taula 4 Valoració economica del material per desenvolupar el prototip

Pel que fa a la valoració en hores del projecte, es calcula les hores invertides en el desenvolupament de l'aplicació des de zero i la instal·lació de cables necessaris per fer arribar les entrades digitals a la targeta Arduino i les proves necessàries per a comprovar que funcioni per deixar el projecte com a finalitzat.

En cas de voler realitzar millores o afegits com a la introducció de sensors com per exemple de temperatura o humitat les modificacions en la programació caldria un nombre molt menor d'hores, ja que s'aprofitaria les funcions desenvolupades per a les noves funcionalitats.

Component	Hores	Preu/Hora	Cost de material
Desenvolupament programari	20	35	700,00 €
Tecnic electric (derivació cables fins a lloc ubicació Arduino)	5	30	150,00 €
Connexió cables Entrada/Sortida a Arduino i instal·lació Arduino	2	30	60,00 €
Verificació i proves insitu	2	30	60,00 €
		TOTAL	970,00 €

Taula 5 Valoració economica del temps invertit per a realitzar el projecte.

3. Conclusions

En primer lloc la realització d'aquest projecte m'ha permès introduir-me en el coneixement d'una plataforma amb múltiples possibilitats i que em podran ser útils de cara a un futur immediat de cara a possibles solucions per monitorar de l'entorn mitjançant sensors de temperatura, humitat, efecte Hall, etc. He pogut determinar que emprant un sistema similar a l'emprat en aquest projecte combinant sensors analògics amb el Shield Ethernet, permet generar una pàgina web on periòdicament ens mostri per exemple la temperatura d'un CPD i aquesta pàgina sigui llegida per un sistema de monitoratge com ara NAGIOS, el qual en pugui donar una gràfica de l'evolució de la temperatura del CPD. Aquesta informació és molt valuosa per qualsevol administrador de sistemes.

Pel que fa als objectius plantejats, considero que en el prototip s'han resolt el 90%. El 10% restant considero que ve donat per la incapacitat que m'he trobat a l'hora de poder calibrar adequadament els sensors MQ7, en el "datasheet" del producte ens indica una gràfica on es relaciona la concentració de gasos en PPM amb la resistència variable generada dividida per una constant que és la R_0 (resistència obtinguda quan la concentració de CO en un ambient de 100 PPM) . Aquest sensor té una sensibilitat entre 20 i 10000 PPM però en l'atmosfera hi ha de fet una concentració de 0,1 PPM. Caldria un laboratori per poder fer l'obtenció de valor R_0 en un ambient controlant amb uns valors PPM coneguts.

A causa del desconeixement de la plataforma i amb el treball amb sensors analògics la planificació del projecte ha sofert algunes desviacions tant en l'àmbit positiu com negatiu. Gràcies a la quantitat d'informació que hi ha a Internet respecte a la plataforma d'Arduino, als exemples de diferents projectes subministrats per la comunitat d'usuaris i a la similitud del llenguatge de programació amb el llenguatge C+, la corba d'aprenentatge ha estat més ràpid que l'esperat. M'he trobat problemes inesperats a l'hora d'optimitzar el llenguatge a fi de disminuir la càrrega de variables sobre la memòria RAM i sobretot amb el calibratge del sensor MQ7.

Malgrat que els problemes esmentats ha permès un millor coneixement del treball amb sensors analògics que em permetrà en un futur treballar amb aquest tipus de dispositius per a realitzar captures de diferents valors ambientals.

De cara a un futur, i emprant sistema amb més potencia (com ara Arduino Mega o similars) es podria implementar sistemes de control de domòtica, com ara control de llums encesos, sistema d'obertura de persianes, calefacció, etc. Tot i que es troba a faltar en Arduino un sistema de multitasca que hi ha en altres plataformes de microcontroladors, ja que en la plataforma emprada

només disposa d'un fil d'execució encara que aquest desavantatge ho compensa amb la facilitat de programació possibilita fer el salt cap a altres plataformes més potents.

4. Glossari

Arduino: Plataforma d'Hardware lliure destinada a facilitar l'ús de l'electrònica i la programació de sistemes encastats per diferents projectes per diferents àrees. Tant el Hardware com el Software de la interfície de desenvolupament són de codi obert.

Biblioteca: Conjunt d'implementacions definides a un llenguatge informàtic per a la realització de tasques concretes (matemàtiques, treballa amb cadena de caràcters, etc) que poden ser invocades per a poder utilitzar els mètodes o funcions que conte a fi de facilitar la tasca al programador .

Diagrama de Gantt: Representació gràfica per mostrar la planificació temporal de un projecte

Open-source: Sistema obert que pot ser utilitzar de forma gratuïta i modificable per part de l' usuari, es pot trobar sistemes Open-Source tant a nivell de Hardware com de Software.

PPM : Parts Per Milió, indica la concentració d' una gas en

Relay o Relé: Dispositiu que como un interruptor controlat per un circuit elèctric en que per mitja d' una bobina i un electroimant s' acciona un joc de un o més contactes per permeten obrir o tancar altres circuits elèctrics independents

LED : Acrònim en angles de Light Emiting Diode, es tracta d' un tipus de díode que emet llum. Inicialment s' utilitzaven com indicadors en dispositius electrònics

Shield: Dispositiu en forma de placa que permet connectar-se acopant-se el connector que aporta al pins de l' Arduino i permet estendre la seves funcionalitats (xarxa, WiFi, relé, etc)

UML: *Unified Modeling Language*. Llenguatge de modelat que permet descriure gràficament los mètodes o processos de un sistema i com es interrelacionen entre ells.

SMTP: Send Mail Transfer Protocol, Protocol destinat a l'enviament de correu i que esta definit en els RFC 821 (1982) , RFC 2821 (2001) i RFC 5321 (2008)

5. Bibliografia

- [1] <http://arduino.cc/es/Tutorial/Sketch> [visitat Octubre 2016]
- [2] http://www.naylampmechatronics.com/blog/42_Tutorial-sensores-de-gas-MQ2-MQ3-MQ7-y-MQ13.html [visitat Novembre 2016]
- [3] <https://www.sparkfun.com/products/9403> [visitat Desembre 2016 2014]
- [4] <https://temboo.com/> [visitat Novembre 2014]
- [5] <https://es.wikipedia.org/wiki/Base64> [visitat Novembre 2016]
- [6] http://www.seeedstudio.com/wiki/Relay_Shield_V2.0 [visitat Octubre 2016]
- [7] <http://thesis.jmsaavedra.com/prototypes/software/mq-7-breakout-arduino-library/> [visitat Octubre 2016]
- [8] <http://fritzing.org/home/> [visitat Novembre 2016]
- [9] <http://forum.arduino.cc/index.php?topic=55799.msg399839#msg399839> [visitat Novembre 2016]
- [10] <http://www.murciasalud.es/pagina.php?id=180398&idsec=1573> [visitat Novembre 2016]
- [11] <http://sinclair.carm.es/calidadaire> [visitat Novembre 2016]
- [12] <http://diymakers.es/crear-servidor-web-con-arduino/> [visitat Octubre 2016]
- [13] <https://www.arduino.cc/en/Tutorial/WebServer> [visitat Octubre 2016]
- [14] <http://www.luisllamas.es/2014/09/leer-un-pulsador-con-arduino/> [visitat Setembre 2016]
- [15] <http://playground.arduino.cc/CommonTopics/PullUpDownResistor> [visitat Setembre 2016]

6. Annexos

6.1 Diagrama d'estats

Diagrama d'estats del programa en general

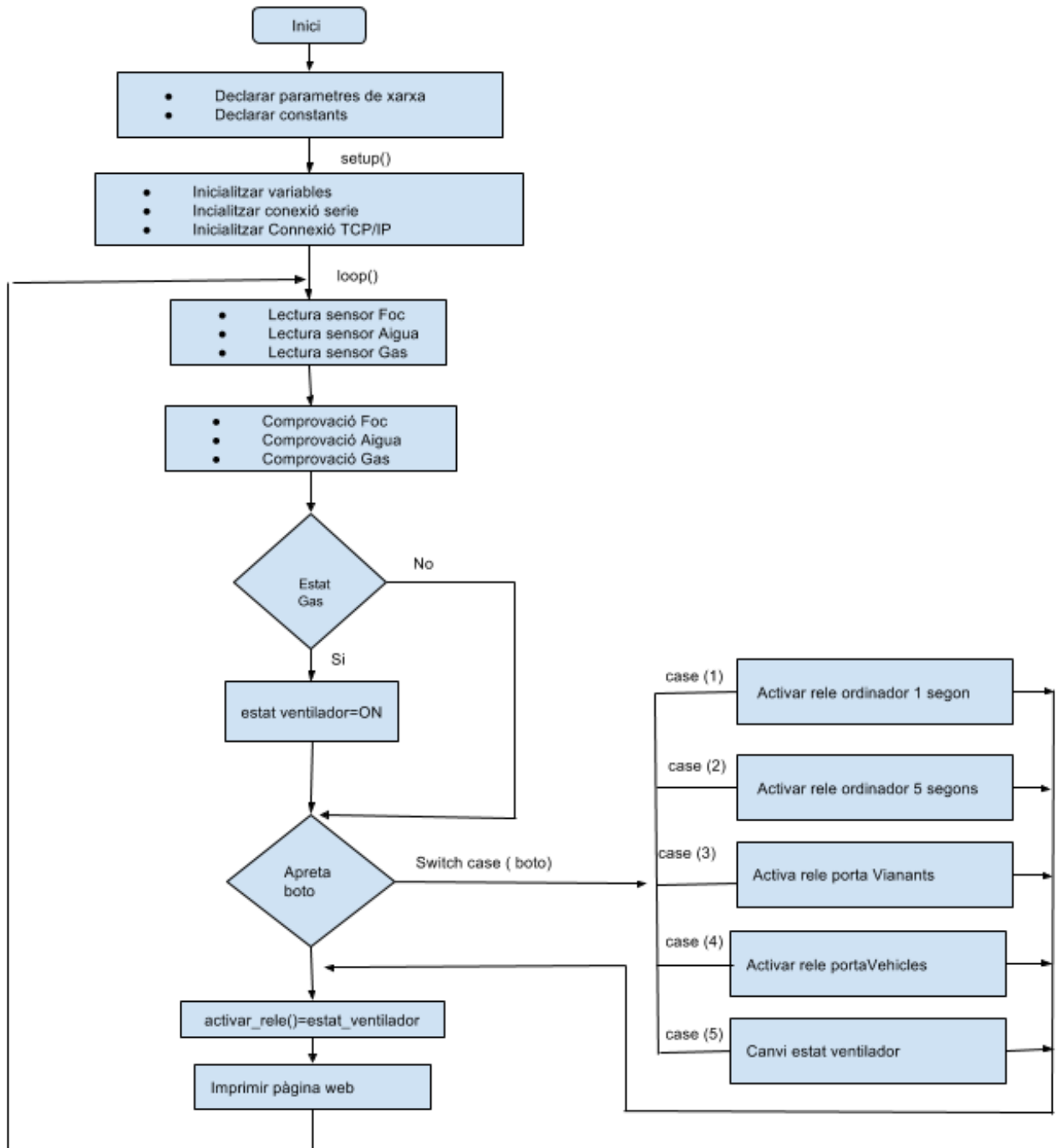
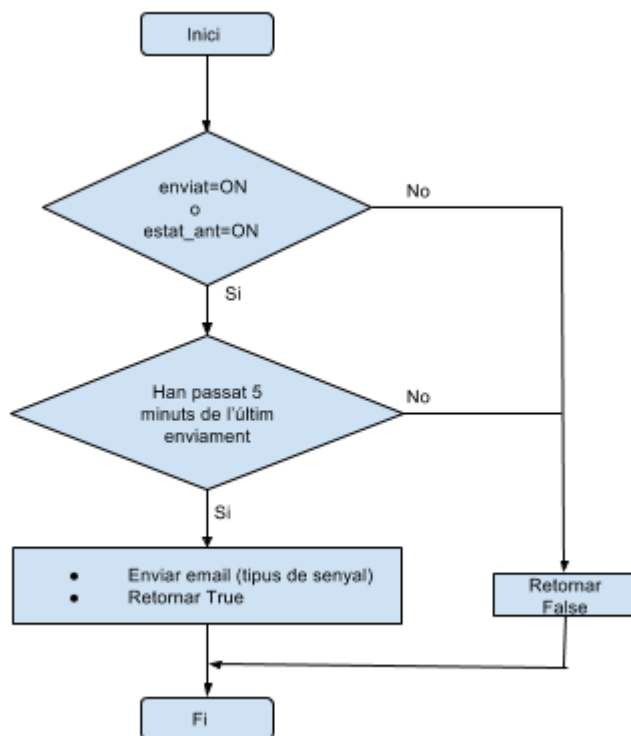


Diagrama d'estats de control d'estat (enviament) :

Aquesta funció segons diferents paràmetres d'entrada si s'ha d'enviar o no el correu electrònic d'avis. El valor que ens retornarà a més indicarà si s'ha enviat o no el correu.

Com a variables d' entrada hi ha :

- Estat anterior: Ens indica l'estat que hi havia en l'anterior cicle.
- Enviat: Ens indica si s'ha enviat el correu electrònic
- Temps: Indica el moment en que es va enviar el correu electrònic, aquest valor són el milisegons des de l'ultima vegada que es va iniciar la placa Arduino.
- Enviament: Indica el tipus de variable que controlen (FOC, AIGUA o GAS)



6.2 Codi Font:

Amb la documentació de la memòria s'adjunta el codi font que es va utilitzar en la PAC2 per a fer el vídeo amb el prototip en execució. Codi_pac2.ino

També s'adjunta el codi amb les modificacions realitzades entre l'entrega de la PAC2 i la de la PAC3: Cod_pac3.ino

Les diferències entre els dos codis són els següents:

- Una millora en l'assignació de variables a fi d'optimitzar el seu espai (emprant bytes enlloc d'INT quan convingui o passant d'INT. a boolean)
- Eliminar gran part de codi repetit en el bucle principal per posar-ho com a funció que es crida en aquest bucle.
- Afegir el paràmetre F, en les instruccions que contenen el mètode println o print a fi de traslladar el contingut de la memòria principal a la memòria Flash alliberant memòria RAM.

Aquest darrer codi font no està del tot depurat en haver-se cremat part de la placa per culpa d'un curtcircuit afectant els pins digitals d'entrada i s'ha adquirit de cara a la conclusió del projecte una placa Arduino UNO i un Shield Ethernet per suplir el mòdul Arduino Ethernet malmès..

Codi Font Final.

```
1. #include <SPI.h>
2. #include <Ethernet2.h>
3. byte mac[] = {0x90,0xA2,0xDA,0x10,0xC5,0x80};
4. //MAC
5. IPAddress ip(192,168,1,177);
6. //IP
7. EthernetServer servidor(8080);
8. IPAddress gateway( 192, 168, 1 , 1 );
9. IPAddress subnet( 255, 255, 255, 0 );
10. // Adreça IP del servidor de correu SMTP que emprarem
11. IPAddress server( 1, 1, 1,1 ); /
12. EthernetClient client_smtp;
13. const byte PIN_aigua=3;
14. const byte PIN_foc=8;
15. const byte PIN_rele1=4;
16. const byte PIN_rele2=5;
17. const byte PIN_rele3=6;
18. const byte PIN_rele4=7;
19.
20. const boolean ON=true;
21. const boolean OFF=false;
22. const int Freqüencia_enviament_mail=30000; //establerta en milisegons
23. boolean state_foc;
24. boolean state_aigua;
```

```

25. boolean state_anterior_aigua;
26. boolean state_anterior_foc;
27. boolean state_aigua_enviat;
28. boolean state_foc_enviat;
29. boolean state_gas_enviat;
30. boolean state_ventilador;
31. boolean state_gas;
32. boolean state_anterior_gas;
33. //boolean canvi;
34. int valor;
35. boolean enviat_aigua=false;
36. boolean enviat_foc=false;
37. boolean enviat_gas=false;
38. String enviament=String(5);
39. int temps_aigua_si;
40. long temps_foc_si;
41. long temps_aigua;
42. long temps_foc;
43. long temps_gas;
44. long temps_control;
45. void setup() {
46.
47. //Definim l'estat dels pins de entrada digitals per connectar l'estat foc i
   aigua
48. pinMode(PIN_aigua,INPUT);
49. pinMode(PIN_foc,INPUT);
50.
51. //Definim l'estat dels pins de sortida per connectar els relés
52. pinMode(PIN_rele1, OUTPUT);
53. pinMode(PIN_rele2, OUTPUT);
54. pinMode(PIN_rele3, OUTPUT);
55. pinMode(PIN_rele4, OUTPUT);
56.
57. //variables d'estat
58. state_anterior_foc=OFF;
59. state_foc=OFF;
60. state_anterior_aigua=OFF;
61. state_aigua=OFF;
62. state_ventilador=OFF;
63. state_anterior_gas=OFF;
64. state_aigua_enviat=state_aigua;
65. state_foc_enviat=state_foc;
66. //carregem el port serie a 9600 bps
67. Serial.begin(9600);
68. Serial.println(F("Inici de la aplicacio"));
69. //carreguem els servidor http
70. Ethernet.begin(mac, ip);
71. servidor.begin();
72. }
73.
74. void loop() {
75.
76.
77. //*****Lectura MQ-7
78. boolean state_gas=llegir_Analog_MQ7();
79.
80. //Lectura Pin digital FOC I AIGUA
81. boolean state_foc=llegir_digital(PIN_foc);
82. boolean state_aigua=llegir_digital(PIN_aigua);
83. if (state_gas) {state_ventilador=ON;} //en cas que es detecti gas activarem
   el ventilador
84. //*****control canvi
85. if (canvi_estat(state_foc,state_anterior_foc)) {
86. temps_foc=millis();
87. enviat_foc=false;

```

```

88.  }
89.  else
90.  {
91.    enviat_foc=control_estat (state_foc,state_anterior_foc,2,temps_foc,enviat_
foc,state_foc_enviat);
92.    if (enviat_foc) {
93.      state_foc_enviat=state_anterior_foc;
94.      temps_foc=millis();
95.    }
96.  }
97.
98.  if (canvi_estat(state_aigua,state_anterior_aigua)) {
99.    temps_aigua=millis();
100.    enviat_aigua=false;
101.  }
102.  else
103.  {
104.    enviat_aigua=control_estat (state_aigua,state_anterior_aigua,1,temp
s_aigua,enviat_aigua,state_aigua_enviat);
105.    if (enviat_aigua) {
106.      state_aigua_enviat=state_anterior_aigua;
107.      temps_aigua=millis();
108.    }
109.  }
110.
111.
112.    if (canvi_estat(state_gas,state_anterior_gas)) {
113.      temps_gas=millis();
114.      enviat_gas=false;
115.    }
116.    else
117.    {
118.      enviat_gas=control_estat (state_gas,state_anterior_gas,3,temps_gas,
enviat_gas,state_gas_enviat);
119.      if (enviat_gas) {
120.        state_gas_enviat=state_anterior_gas;
121.        temps_gas=millis();
122.      }
123.    }
124.
125.
126.    state_anterior_gas=state_gas;
127.    state_anterior_aigua=state_aigua;
128.    state_anterior_foc=state_foc;
129.
130.    if (state_ventilador)
131.    {
132.      digitalWrite(7,HIGH);
133.    }
134.    else
135.    {
136.      digitalWrite(7,LOW);
137.    }
138.    //***** part de generació HTML i gest
io de botons**
139.    //*****
*****
140.    EthernetClient client = servidor.available();
141.    if (client) {
142.      boolean lineaActual = true;
143.      String cadena="";
144.      while (client.connected()) {
145.        if (client.available()) {
146.          char c = client.read();
147.          // Serial.write(c);

```

```

148.         cadena.concat(c);
149.         int posicion=cadena.indexOf("BOTO="); //
150.         if(cadena.substring(posicion)=="BOTO=01")
151.         {
152.             digitalWrite(4,HIGH);
153.             delay(1000);
154.             digitalWrite(4,LOW);
155.         }
156.         if(cadena.substring(posicion)=="BOTO=02")
157.         {
158.             digitalWrite(4,HIGH);
159.
160.             delay(5000);
161.             digitalWrite(4,LOW);
162.         }
163.         if(cadena.substring(posicion)=="BOTO=03")
164.         {
165.             digitalWrite(5,HIGH);
166.
167.             delay(1000);
168.             digitalWrite(5,LOW);
169.         }
170.
171.         if(cadena.substring(posicion)=="BOTO=04")
172.         {
173.             digitalWrite(6,HIGH);
174.             delay(1000);
175.             digitalWrite(6,LOW);
176.         }
177.         if(cadena.substring(posicion)=="BOTO=05")
178.         {
179.             if (state_ventilador) { state_ventilador=OFF;}
180.             else
181.             {
182.                 state_ventilador=ON;
183.             }
184.         }
185.         if (c == '\n' && lineaActual) {
186.             client.println("HTTP/1.1 200 OK");
187.             client.println("Content-Type: text/html");
188.             client.println();
189.             client.println(F("<html>"));
190.             client.println(F("<head>"));
191.             client.println("<meta http-
equiv=\"refresh\" content=\"5;URL=http://192.168.1.177:8080\">");
192.             client.println(F("</head>"));
193.             client.println(F("<body>"));
194.             client.println(F("<h1 align='center'>CONTROL </h1><h3 align='
center'>PARKING</h3>"));
195.             client.println(F("<div style='text-align:center;'>"));
196.             client.print("<br>");
197.             if (state_foc==ON) {
198.                 client.println(" <font color=\"red\">Estat del alarma de fo
c: Alarma Activada");
199.                 client.println("<font>");
200.             } else {
201.                 client.println(" <font color=\"green\">Estat del alarma de
foc: Correcte ");
202.                 client.print("<font>");
203.             }
204.             client.print("<br><br>");
205.             if (state_aigua==ON) {
206.                 client.println(" <font color=\"red\">Estat del alarma aigua
: Alarma Activada");
207.                 client.println("<font>");

```



```

208.         } else {
209.             client.println(" <font color=\"green\">Estat del alarma de
aigua: Correcte ");
210.             client.println("<font>");
211.         }
212.         client.println("<br><br>");
213.         if (state_gas) {
214.             client.println(" <font color=\"red\">Estat del alarma de Ga
s: Alarma Activada");
215.             client.println("<font>");
216.         } else {
217.             client.println(" <font color=\"green\">Estat del alarma de
Gas: Correcte ");
218.             client.print("<font>");
219.         }
220.         client.print("<br><br>");
221.         client.println(F("<button onClick=location.href='./?BOTO=01\'
style='margin:auto;background-
color: #85aaea;color: snow;padding: 10px;border: 1px solid #3F7CFF;width:200px
;'>"));
222.             client.println(F("Engegar Ordinador"));
223.             client.println(F("</button>"));
224.             client.println(F("<button onClick=location.href='./?BOTO=02\'
style='margin:auto;background-
color: #85aaea;color: snow;padding: 10px;border: 1px solid #3F7CFF;width:200px
;'>"));
225.             client.println(F("Aturar Ordinador"));
226.             client.println(F("</button>"));
227.             client.println(F("<br /><br />"));
228.             client.println(F("<button onClick=location.href='./?BOTO=03\'
style='margin:auto;background-
color: #85aaea;color: snow;padding: 10px;border: 1px solid #3F7CFF;width:200px
;'>"));
229.             client.println(F("Obrir Porta de vianants"));
230.             client.println(F("</button>"));
231.             client.println(F("<button onClick=location.href='./?BOTO=04\'
style='margin:auto;background-
color: #85aaea;color: snow;padding: 10px;border: 1px solid #3F7CFF;width:200px
;'>"));
232.             client.println(F("Obrir Porta de Vehicles"));
233.             client.println(F("</button>"));
234.             client.println(F("<br /><br />"));
235.
236.             if (state_ventilador==ON){
237.                 client.println(F("<button onClick=location.href='./?BOTO=05
\' style='margin:auto;background-
color: #ea8f85;color: snow;padding: 10px;border: 1px solid #3F7CFF;width:200px
;'>"));
238.
239.                 client.println(F("apagar vent"));
240.             }
241.             else
242.             {
243.                 client.println(F("<button onClick=location.href='./?BOTO=05
\' style='margin:auto;background-
color: #85ea8e;color: black;padding: 10px;border: 1px solid #3F7CFF;width:200p
x;'>"));
244.
245.                 client.println(F("engegar vent"));
246.             }
247.             client.println(F("</b><br />"));
248.             client.println(F("</b></body>"));
249.             client.println(F("</html>"));
250.             break;
251.         }

```

```

252.         if (c == '\n') {
253.             lineaActual = true;
254.         }
255.         else if (c != '\r') {
256.             lineaActual = false;
257.         }
258.     }
259. }
260.     delay(1);
261.     client.stop();
262. }
263. }
264.
265. //*****
266. boolean llegir_digital (int pin){
267.     //per fer la lectura del pin digital llegirem 10 vegades el valor i e
268.     l valor que més vegades es llegeixi serà marcarà l'estat
269.     int valor=0;
270.     int buttonState=0;
271.     for (int i=0;i<10;i++)
272.     {
273.         buttonState = digitalRead(pin);
274.         if (buttonState == HIGH) {valor++;}
275.         delay(10);
276.     }
277.     if (valor>4) {return true;}
278.     return false;
279. }
280.
281. //*****
282.
283. byte sendEmail(int enviament,boolean state ) {
284.     byte thisByte = 0;
285.     byte respCode;
286.     if(client_smtp.connect(server,25)) {
287.         Serial.println(F("connected"));
288.     } else {
289.         Serial.println(F("connection failed"));
290.         return 0;
291.     }
292.     if(!eRcv()) return 0;
293.     Serial.println(F("Sending helo"));
294.     // change to your public ip
295.     client_smtp.println(F("helo a.com"));
296.     if(!eRcv()) return 0;
297.     client_smtp.println("AUTH LOGIN");
298.     if(!eRcv()) return 0;
299.     //Enviem usuario en base64
300.     client_smtp.println("11111111111111==");
301.     if(!eRcv()) return 0;
302.     //Enviem el password en base64
303.     client_smtp.println("2222222222222222=");
304.     if(!eRcv()) return 0;
305.     Serial.println(F("Sending From"));
306.     // enviem l'adreça del remitent
307.     client_smtp.println(F("MAIL From: alarmes@uoc.com"));
308.     if(!eRcv()) return 0;
309.     Serial.println(F("Sending To"));
310.     client_smtp.println(F("RCPT To: notificacions@gmail.com"));
311.     if(!eRcv()) return 0;
312.     Serial.println(F("Sending DATA"));
313.     client_smtp.println(F("DATA"));
314.     if(!eRcv()) return 0;
315.     Serial.println(F("Sending email"));

```

```

316.     client_smtp.println(F("To: dflorga <notificacions@gmail.com>"));
317.     client_smtp.println(F("From: Alarmes Arduino <notificacions@uoc.com>"
));
318.     Serial.print("enviament es ");
319.     Serial.println(enviament);
320.     switch (enviament) {
321.     case 1:
322.     {
323.         Serial.println("enviem mail d'aigua");
324.         if (state==OFF) {
325.             client_smtp.println(F("Subject: Alarma Aigua estat OK\r\n"));
326.             client_smtp.println(F("Estat de aigua correcte!"));
327.             Serial.println(F("Subject: Alarma Aigua estat OK\r\n"));
328.             Serial.println(F("Estat de aigua correcte!"));
329.
330.         } else {
331.             client_smtp.println(F("Subject: ALARMA AIGUA!!!!!!!!!!!! \r\n"));
332.             client_smtp.println(F("DETECTADA PRESSIO DE AIGUA INCORRECTE!"));
333.             Serial.println(F("Subject: ALARMA AIGUA!!!!!!!!!!!! \r\n"));
334.             Serial.println(F("DETECTADA PRESSIO DE AIGUA INCORRECTE!"));
335.         }
336.
337.
338.     }
339.     break;
340.     case 2:
341.     {
342.         Serial.println("enviem mail de foc");
343.         if (state==OFF) {
344.             client_smtp.println(F("Subject: Alarma FOC estat OK\r\n"));
345.             client_smtp.println(F("Estat de Alarma de foc correcte!"));
346.             Serial.println(F("Subject: Alarma FOC estat OK\r\n"));
347.             Serial.println(F("Estat de Alarma de foc correcte!"));
348.         } else {
349.             client_smtp.println(F("Subject: ALARMA FOC!!!!!!!!!!!! \r\n"));
350.             client_smtp.println(F("DETECTADA ALARMA INCENDIS ACTIVADA!"));
351.             Serial.println(F("Subject: ALARMA FOC!!!!!!!!!!!! \r\n"));
352.             Serial.println(F("DETECTADA ALARMA DE FOCS ACTIVADA!"));
353.         }
354.     }
355.     break;
356.     case 3:
357.     {
358.         if (state==OFF) {
359.             client_smtp.println(F("Subject: Alarma GAS estat OK\r\n"));
360.             client_smtp.println(F("Estat de Alarma de GAS correcte!"));
361.             Serial.println(F("Subject: Alarma GAS estat OK\r\n"));
362.             Serial.println(F("Estat de Alarma de foc correcte!"));
363.         } else {
364.             client_smtp.println(F("Subject: ALARMA GAS!!!!!!!!!!!! \r\n"));
365.             client_smtp.println(F("DETECTADA ALARMA GAS ACTIVADA!"));
366.             Serial.println(F("Subject: ALARMA GAS!!!!!!!!!!!! \r\n"));
367.             Serial.println(F("DETECTADA EXCES DE MONOXID DE CARBONI!"));
368.         }
369.     }
370.     break;
371. }
372. client_smtp.println(F("."));
373. if(!eRcv()) return 0;
374. Serial.println(F("Sending QUIT"));

```

```

375.     client_smtp.println(F("QUIT"));
376.     if(!eRcv()) return 0;
377.     client_smtp.stop();
378.     Serial.println(F("disconnected"));
379.     return 1;
380. }
381.
382. //*****
383. byte eRcv() {
384.     byte respCode;
385.     byte thisByte;
386.     int loopCount = 0;
387.     while(!client_smtp.available()) {
388.         delay(1);
389.         loopCount++;
390.         if(loopCount > 10000) {
391.             client_smtp.stop();
392.             Serial.println(F("\r\nTimeout"));
393.             return 0;
394.         }
395.     }
396.     respCode = client_smtp.peek();
397.     while(client_smtp.available()) {
398.         thisByte = client_smtp.read();
399.         Serial.write(thisByte);
400.     }
401.     if(respCode >= '4') {
402.         efail();
403.         return 0;
404.     }
405.     return 1;
406. }
407.
408. //*****
409. void efail() {
410.     byte thisByte = 0;
411.     int loopCount = 0;
412.     client_smtp.println(F("QUIT"));
413.     while(!client_smtp.available()) {
414.         delay(1);
415.         loopCount++;
416.         if(loopCount > 10000) {
417.             client_smtp.stop();
418.             Serial.println(F("\r\nTimeout"));
419.             return;
420.         }
421.     }
422.     while(client_smtp.available()) {
423.         thisByte = client_smtp.read();
424.         Serial.write(thisByte);
425.     }
426.     client_smtp.stop();
427.     Serial.println(F("disconnected"));
428. }
429.
430. //*****
431. boolean Llegir_Analog_MQ7() {
432.     int MQ7 = analogRead(A0);
433.     delay(100);
434.     if (MQ7>=100) {
435.         return ON ;
436.     }
437.     return OFF;
438. }
439.

```

```

440. //*****
441. boolean control_estat (boolean estat_actual,boolean estat_anterior, int
enviament, long temps, boolean enviat, boolean estat_enviat)
442. {
443.
444.     if (!(enviat && estat_anterior==OFF)) {
445.         // versuio optimitzada de if ((no enviat && state_anterior=="OFF")
|| (state_anterior=="ON")){
446.
447.         if (estat_enviat==ON || estat_anterior==ON) {
448.             long temps_control=millis()-temps;
449.
450.             if (temps_control>Frequencia_enviament_mail) {
451.                 if(sendEmail(enviament,estat_actual)) Serial.println(F("Email se
nt")); else Serial.println(F("Email failed"));
452.                 Serial.print(F("enviem mail , enviat a les "));
453.                 Serial.print(millis());
454.                 if (estat_anterior==OFF) {
455.                     Serial.println(F(" no hi haura proper mail a no ser que es ca
nvii d'estat "));
456.                 } else {
457.                     Serial.print(F(" el proper sera a les "));
458.                     Serial.println(millis()+Frequencia_enviament_mail);
459.                 }
460.                 delay(100);
461.                 return true;
462.             }
463.         }
464.     }
465.     return false;
466. }
467. //*****
468. boolean canvi_estat (boolean estat_anterior, boolean estat_actual)
469. {
470.     return !(estat_anterior==estat_actual) ;
471. }

```