Projecte final de curs

Advanced Data logger for industrial dust collector based on Raspberry Pi board.

Projectista: Màxim Horcajo Herrera
Tutor: Juan José Alins Delgado
President: José Luis Muñoz Tapia
Vocal: Jorge Martin Gimenez

Advanced Data logger for industrial dust collector based on Raspberry Pi board.

Index

1. INTRODUCCIÓ

1.1. Motivació

En l'actualitat, l'adquisició de dades en temps real i l'enviament d'aquestes a un nucli per ser processades conjuntament amb moles altres dades de procedències diverses, forma part d'una tendència en expansió tant en en el món industrial com en el domèstic. Els baixos costs i les dimensions reduïdes dels nous miniordinadors com la Rasberrypi permeten una implantació més senzilla i viable. Per aquests motius aquest tipus de sistemes són utilitzats tant per estudiants inexperts com per a enginyers.
Aquest projecte em va interessar per la creixent demanda i ús d'aquest tipus de sistemes, i perquè és un projecte que requereix de diferents matèries com: electrònica, telemàtica i programació cosa que em permetrà ampliar els coneixements en aquests camps tant fonamentals per el futur digital.

1.2. Descripció

En aquest projecte es mostrarà com configurar un ordinador de placa reduïda com la Raspberry Pi amb el programari adient per a que funcioni com a una unitat avançada d'adquisició i emmagatzematge de dades d'uns sensors situats dins d'un sistema captador de pols industrial AE-9 de l'empresa FAPP. El sistema ha de gestionar els senyals dels sensors i notificar en cas d'incidències, per aquesta tasca s'utilitzarà el programari del pack OMD (Open Monitoring Distribution). El data-logger dissenyat en aquest projecte està pensat per a que tant el fabricant com el client puguin accedir a les dades d'aquests sensors sempre que sigui necessari, i a un cost més baix possible. Aquest disseny és un exemple de molts altres que també es poden aplicar a un sistema programable per xarxa a una màquina de qualsevol tipus i característiques.

1.3 Disseny

Un enregistrador de dades, en anglès s'anomena "Data Logger", que és un sistema electrònic per a mesurar un cert nombre de variables i efectuar una tabulació escrita i/o registrar-les en un format adequat per a l'entrada a l'ordinador. El preu dels enregistradors de dades ha anat disminuint al llarg dels anys i generalment són petits, accionats amb piles, portàtils i equipats en un microprocessador, amb memòria interna per emmagatzemar dades dels sensors. Alguns fan interfície amb un ordinador personal i fan servir programari per activar l'enregistrador de dades, veure i analitzar les dades recollides, mentre altres tenen un dispositiu d'interfície local (keypad, LCD) i poden ser usats com un dispositiu autònom.
Un dels beneficis principals de fer servir enregistradors de dades és la possibilitat de recollir dades automàticament les 24 hores del dia.

El captador de pols AE-9 de FAPP és un aspirador estacionari utilitzat per a la captació de pols de cabal mig. En funció del tipus de pols a aspirar, el AE-9 s'adapta tant en superfície filtrant, potència, requeriments ATEX i dipòsits suplementaris. Està fabricat amb INOX i està orientat sobretot cap al sector alimentari i farmacèutic.

Les seves característiques són:

Alimentació	230/400 V
Potència	> 3 kW
Aspiració	370 mm/H2O
Cabal d'aire	5.000 m3/h
Dipòsit	100 l.
Dimensions (llarg x alt x amplada)	80 x 268 x 80 cm.
Pes	380 Kg.
Nivell sonor	72 dB (A)

Taula 1. Característiques AE-9

Imatge 1. Captador de pols AE-9

Es connectaran a través d'una targeta perifèrica amb DAC (Digital/Analog Converter) per a sensors de bucle de corrent de 4 a 20 mA i que es connecta en els busos I2C de la RaspberryPi. Requereix programació en C per a comunicar-se a través dels busos I²C.

Imatge 2. Un sensor de pressió

No ha sigut possible provar empíricament les dades recollides per sensors en un captador de pols original, i s'ha considerat convenient simular les dades dels sensors de potència, cabal d'aire, capacitat del dipòsit i la pressió que té l'aire aspirat, per a tenir valors diferents i tenir una idea del resultat final. Simularem els sensors de dues maneres diferents:

- Simulació amb una font d'alimentació:

  La lectura de valors de corrent es simularà a partir de fonts d'alimentació variables, de l'aula de Projectes. Aquests valors són llegits pel programa en C a través de la placa I²C, sent la prova més fiable de que el sistema funciona.

- Simulació per programació Script:

  Aquest tipus de simulació és el que més s'ha utilitzat per a fer proves, ja que es poden tenir varis sensors simulats a l'hora, mentre que a través de la placa I²C només hi han quatre entrades. S'exageraran els valors utilitzats per a tenir rangs més elevats.

El pack de software que es farà servir, OMD (Open Monitoring Distribution), conté el programari del Nagios, que realitza comprovacions periòdiques, facilita la gestió de les dades, i genera notificacions per SMS o correu electrònic. Es crearan els plugins necessaris per habilitar les notificacions per a encendre leds d'una altre placa externa, anomenada Piface Digital.
Finalment, s'utilitzarà un altre programa, anomenat Thruk, que l'utilitzarem per a poder consultar les dades en un navegador d'Internet.

A més a més, el sistema dissenyat en aquest projecte, esta pensat per a ser configurat a través de tres tipus de connexions diferents, segons les necessitats requerides;

-Xarxa a través de connexió per cable Ethernet.
-Xarxa Wifi o servidor Access Point a través de connexió USB.
-Xarxa 3G a través de connexió USB amb un mòdem de de la companyia Vodafone.

Es farà servir una Xarxa Virtual Privada (VPN) per a assegurar l'accés segur.

Tot el material fet servir ha sigut lliurat pel tutor del projecte i pertany al departament de Telemàtica de la UPC EET, excepte el material utilitzat a l'aula de Projectes.

Per a entendre tots els conceptes que s'utilitzen i abans de començar a explicar perquè està dissenyat i configurat el nostre sistema d'aquesta manera, es farà una descripció detallada de la majoria de la informació mostrada.

2. TIPUS DE CONNECTIVITATS

Per a poder entendre certs apartats d'aquest projecte es considera necessari explicar informació relativa al projecte, començant per casi tots els protocols utilitzats.
En una comunicació client-servidor s'entén que el client és qui requereix un servei del servidor a través d'una xarxa privada (LANo WLAN) o una xarxa pública com Internet.

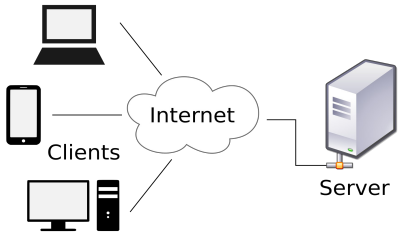


Diagrama 1. Connexió Client-servidor

2.1. Protocols

Els protocols estan presents en totes les etapes necessàries per establir una comunicació client-servidor entre equips, des de les etapes de més baix nivell (p. ex. la transmissió de fluxos de bits a un medi físic) fins a aquelles etapes de més alt nivell (p. ex. el compartir o transferir informació des d'una computadora a una altra dins de la xarxa).

Per començar, s'explicaran els protocols de baix nivell usats per a la comunicació entre plaques de circuits integrats, i darrerament s'explicarà per sobre el funcionament d'Internet a partir del model de referència OSI, i seguidament s'explicaran els protocols fets servir més importants.

2.1.1. I²C i SPI

L'I2C i l'SPI són protocols de comunicacó en sèrie de baix nivell utitlizats per a la comunicació de dades entre circuits integrats. Aquest protocols s'han dissenyat tenint en ment la minimització de els senyals físiques per tal de reduir el cost, complaxitat i tamany necessaris.

Per a connectar la placa o circuit integrat del convertidor dels sensors de bucle de corrent a la Raspberry mitjançant el port GPIO, s'utilitzaran el busos I²C, i per a connectar la PiFace Digital2, es faran servir els busos SPI.

Els ports **GPIO** (**General Purpose Input/Output**, *Entrada/Sortida per a un Ús General*) són un tipus de ports d'entrada/sortida molt utilitzats en el món dels microcontroladors.
Aquestes són entrades i sortides de senyals físiques molt versàtils que poden ser configurades per actuar de diverses maneres segons les necessitats de l'usuari. En la majoria de microcontroladors actuals podem trobar GPIO que es poden configurar com a entrades i sortides digitals, entrades i sortides analògiques o bits de comunicació per els protocols I2C, SPI o UART entre d'altres funcionalitats. La seva aparició és dels principis dels anys 80.

I²C és un bus de comunicacions en sèrie. El seu nom ve d'*Inter-Integrated Circuit* (Circuits Inter-Integrats). La versió 1.0 data de l'any 1992 i la versió 2.1 de l'any 2000, el seu dissenyador és Philips. La velocitat és de 100Kbits per segon en el mode estàndard, encara que també permet velocitats de 3.4 Mbit/s. És un bus molt utilitzat en la indústria, principalment per a comunicar microcontroladors i els seus perifèrics en sistemes integrats (Embedded Systems) i generalitzant més, per a comunicar circuits integrats entre si que normalment residixen en un mateix circuit imprès. La principal característica d'I²C és que utilitza dos línies per a transmetre la informació: una per a les dades i per un altre el senyal de rellotge.

Com que solen comunicar-se circuits en una mateixa placa que comparteixen una mateixa massa, però si es volen utilitzar circuits integrats separats, són necessàries una tercera i quarta línia, la referència (massa) i l'alimentació del circuit integrat(VCC). La comunicació s'anomena màster/slave ja que el màster és qui porta la batuta (SCL) i slave està obligat a seguir les mateixes normes que el màster.

Les línies s'anomenen:
-SDA: dades
-SCL: rellotge
-GND: massa
-Vdd: voltatge

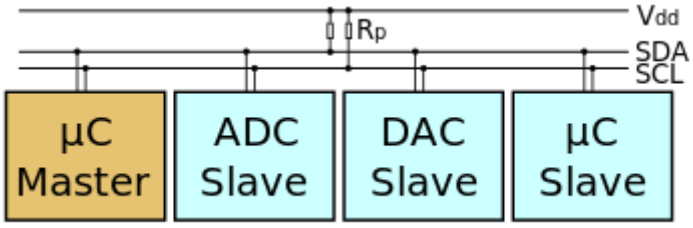


Diagrama 2. Connexió de tres 'Slave' I²C diferents

El bus **SPI** (de l'anglès **Serial Peripheral Interfície**, en català *interfície per a perifèrics sèrie*) és un estàndard de comunicacions, utilitzat principalment per a la transferència de dades entre circuits integrats en equips electrònics. El bus d'interfície de perifèrics sèrie, o bus SPI, és un estàndard per controlar qualsevol dispositiu electrònic digital que accepti un flux de bits sèrie regulat per un rellotge.

El protocol té una línia de rellotge, una línia de dades entrants, una línia de dades sortints i una línia de *xip select*, que activa o desactiva el mòdul de comunicacions del dispositiu amb què desitja comunicar-se. D'aquesta manera, aquest estàndard permet multiplexar les línies del rellotge.

Molts sistemes digitals tenen perifèrics que no necessiten una velocitat de transferència de dades ràpida. Els avantatges d'un bus són que minimitzen el nombre de conductors, pins i la grandària del circuit integrat. Això redueix el cost de fabricar, muntar i provar l'electrònica. Un bus de perifèrics sèrie és l'opció més flexible quan molts tipus diferents d'aquests perifèrics estan presents. La maquinària consisteix en senyals de rellotge, *data in*,*data out* i *xip select* per a cada circuit integrat que ha de ser controlat. Quasi qualsevol dispositiu digital pot ser controlat amb aquesta combinació de senyals. Els dispositius es diferencien en un nombre predictible de formes. Uns llegeixen la dada quan el rellotge puja, d'altres quan el rellotge baixa. Alguns ho llegeixen en el flanc de pujada del rellotge i altres en el flanc de baixada. Escriure és, quasi sempre, en la direcció oposada de la direcció de moviment del rellotge. Alguns dispositius tenen dos rellotges. Un per a capturar o mostrar les dades i l'altre per al dispositiu intern.

El SPI és un protocol síncron. La sincronització i la transmissió de dades entre emissor i receptor (Màster/Slave) es realitza utilitzant aquests 4 senyals:

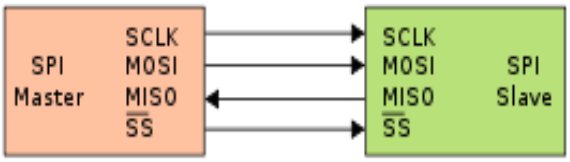


Diagrama 3. Connexió d'un 'Slave' SPI

-SCLK *(Clock)*: És el pols del rellotge que marca la sincronització. A cada pols d'aquest rellotge, es llegeix o s'envia un bit.

-MOSI *(Màster Output Slave Input)*: Sortida de dades del Màster i entrada de dades al Slave.
-MISO *(Màster Input Slave Output)*: Sortida de dades del Slave i entrada al Màster.
-SS/Select: Per seleccionar un Slave, o perquè el Màster li digui al Slave que s'activi.

2.1.2. Model de referència OSI

OSI és el model de referència d'Interconnexió de Sistemes Oberts (OSI) llançat el 1984 i va ser el model de xarxa descriptiu creat per l'ISO (Organització internacional per a l'estandardització). Va proporcionar als fabricants un conjunt d'estàndards que van assegurar una major compatibilitat i interoperabilitat entre els diferents tipus de tecnologia de xarxa produïts per les empreses a escala mundial.

El model de referència OSI s'ha convertit en el model principal per a les comunicacions per xarxa. Encara que existeixen altres models, la majoria dels fabricants de xarxes relacionen els seus productes amb el model de referència de OSI. Aquest model es divideix en set nivells:

Diagrama 4.Model OSI

Nivell 1:

El nivell 1 és la **capa física** i contempla el maquinari de xarxa, el cablejat físic o una connexió Wi-Fi. Gestiona les especificacions elèctriques, el control de col·lisió i d'altres funcions de baix nivell.
El nivell físic s'encarrega de realitzar els serveis sol·licitats pel nivell d'enllaç. No donarem gaire importància a aquest nivell, ja que interessa donar més importància als nivells més elevats.

Nivell 2:

El nivell 2 correspon la **capa d'enllaç de dades** o nivell d'enllaç, s'encarrega de respondre a peticions de servei de la capa de xarxa(nivell 3) i d'entregar les peticions a la capa física (nivell 1).
La capa d'enllaç de dades és la capa encarregada de transferir dades entre nodes de xarxa adjacents en una WAN o entre nodes del mateix segment de LAN. La capa d'enllaç de dades proporciona les funcions i els procediments per a la transmissió de dades entre entitats de xarxa i pot detectar i corregir errors que poden ocórrer a la capa física. Exemples de protocols d'enllaç de dades són Ethernet per a xarxes d'àrea local i PPP, HDLC i ADCCP per a connexions punt a punt.

Nivell 3:

La **capa de xarxa**, també anomenada **capa d'Internet**, es troba entre la Capa d'enllaç de dades i la capa de transport, i s'encarrega d'adreçar missatges i traduir adreces lògiques i noms en adreces físiques. També determina la ruta des de l'origen cap al destí i gestiona problemes de trànsit i controla la congestió de paquets de dades.

Aquesta capa és la que permet als clients posar paquets de dades dins d'alguna xarxa i viatjar als seus destins. És a dir, ofereix el servei d'encaminament dels missatges i de traducció de les adreces lògiques en adreces físiques. També efectua el control de la congestió de la xarxa i la reordenació de paquets un cop arribats a la màquina de destí. És aquí on es defineix un format de paquet oficial i un protocol anomenat IP.

Nivell 4:

La **capa de transport** és l' encarregada de garantir la transmissió de les dades.

La capa de xarxa transfereix datagrames entre dos ordinadors per la xarxa utilitzant com a identificadors les adreces IP. La capa de transport és l'encarregada d'afegir la noció de port. Dins d'una mateixa computadora hi pot haver més d'una aplicació que estigui accedint simultàniament a la xarxa (podem tenir funcionant l'Ares o uTorrent, l'Skipe, el navegador d'Internet amb Facebook, Twiter o la pàgina del correu electrònic,...). Per aquest motiu, quan s'envia un datagrama no en tenim prou amb l'adreça IP de la màquina de destí, necessitem també indicar a quina aplicació estem enviant la informació. Cada aplicació que estigui esperant un missatge utilitzarà un port diferent; estarà a l'espera d'un missatge en un port concret (escoltant un port). S'utilitzarà també un port concret per a l'enviament de missatges.

Els ports tenen una memòria intermèdia (buffer, en anglès) situada entre els programes d'aplicació i la xarxa, de tal manera que les aplicacions transmeten la informació als ports, aquí es van emmagatzemant fins que pugui enviar-se per la xarxa. Un cop transmès arribarà al port destí on s'anirà guardant fins que l'aplicació estigui preparada per a rebre-la.

A més, la capa de transport proporciona un mecanisme per intercanviar les dades entre sistemes finals. El servei de transport orientat a connexió assegura que les dades s'entreguin lliures d'errors, en ordre i sense pèrdues ni duplicacions. És més, la capa de transport pot estar involucrada en l'optimització de l'ús dels serveis de xarxa. És a dir; pot proporcionar la qualitat del servei que s'hagi sol·licitat. Per exemple, l'entitat de sessió pot sol·licitar una tassa d'error determinada, un retard màxim, una prioritat i un nivell de seguretat donat.

Existeixen dos protocols principals dins d'aquesta capa, explicats més endavant:

- TCP (*Transfer Control Protocol*). Ofereix una transferència fiable i orientada a connexió.
- UDP (*User Datagram Protocol*). Of5ereix una transferència no fiable i no orientada a connexió.

Nivell 5:

La **capa de sessió** respon a peticions de servei de la capa de presentació i obté els serveis de la capa de transport.
Aquesta capa ofereix diversos serveis que són crucials per a la comunicació, com són:

1. Control de la sessió a establir entre l'emissor i el receptor (establiment, utilització i alliberament).

2. Control de la concurrència (que dues comunicacions a la mateixa operació crítica no s'efectuïn alhora).

3. Mantenir punts de verificació (*checkpoints*), que serveixen perquè, davant d'una interrupció de la transmissió per qualsevol causa, aquesta es pugui reprendre des de l'últim punt de verificació en comptes de repetir-la des del principi.

Nivell 6:

La **capa de presentació** és responsable del lliurament i formatació d'informació abans d'arribar a la capa d'aplicació per a un posterior processament o per ser mostrada a l'usuari.

Nivell 7:

La **capa d'aplicació** és l'encarregada de fer d'interfície entre l'usuari i la xarxa. Interacciona directament amb els programes d'aplicació dels usuaris, fent ús de protocols d'alt nivell que resolen aspectes de representació, codificació i control de diàleg.

2.1.3. IP

Internet és una xarxa pública i global de ordinadors que estan interconnectats mitjançant el protocol d'Internet i que es comuniquen mitjançant la commutació de paquets.

El **protocol d'Internet** (en anglès *Internet Protocol*), abreviat **IP**, és un protocol no orientat a connexió usat tant per l'origen com per la destinació de la comunicació de dades a través d'una xarxa de paquets commutats no fiable de millor lliurament possible sense garanties.

Les dades en una xarxa basada en IP són enviades en blocs coneguts com a paquets o datagrames (en el protocol IP aquests termes se solen usar indistintament). En particular, en IP no es necessita cap configuració abans que un equip intenti enviar paquets a un altre amb el qual no s'havia comunicat abans.

IP té d'un servei de datagrames no fiable, ho farà el millor possible però garanteixen poc. IP no té cap mecanisme per determinar si un paquet arriba o no al seu destí i únicament proporciona seguretat, ja sigui amb checksums (ACK) o sumes de comprovació, mitjançant les capçaleres i no de les dades transmeses. Per exemple, en no garantir res sobre la recepció del paquet, aquest podria arribar danyat, en un altre ordre pel que fa a altres programes duplicat o simplement no arribar. Si es necessita fiabilitat, aquesta és proporcionada pels protocols de la capa de transport, com TCP.

Una **xarxa privada**, segons la terminologia d'Internet, és una xarxa que utilitza un espai d'adreces IP especificades en el document *RFC 1918*. Es pot assignar adreces d'aquest espai d'adreces als terminals quan cal que aquests es puguin comunicar amb altres terminals dins la xarxa interna, que no forma part d'Internet, però no directament amb Internet.

Les xarxes privades són prou comunes als esquemes de xarxa d'àrea local (LAN) d'oficina, donat que moltes empreses no tenen la necessitat d'una adreça IP global o pública, per a cada estació de treball, impressora o d'altres dispositius dels que disposi l'empresa. Una altra raó per a la utilització de les adreces IP privades és l'escassetat d'adreces IP públiques. Just per evitar aquest problema es va crear l'estàndard IPv6, però encara no s'ha implantat a nivell general.

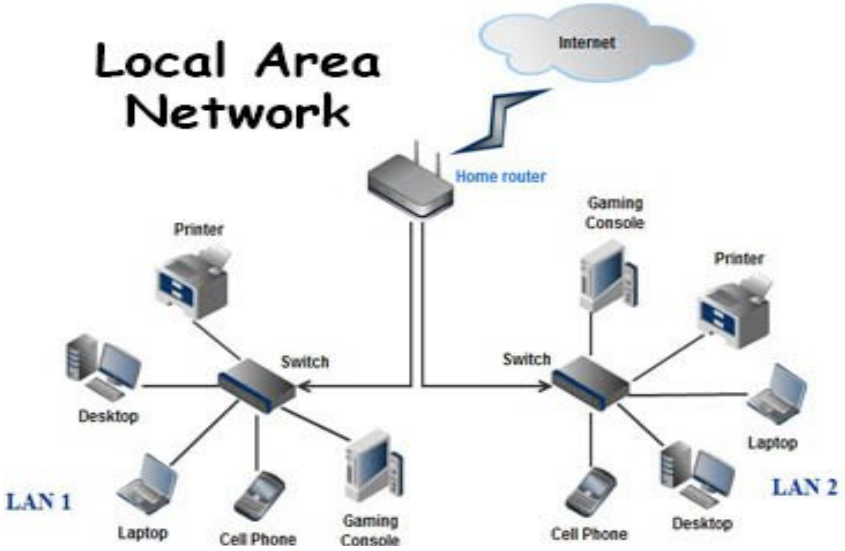


Diagrama 5. Exemple de connexió de dues LAN a un mateix router local.

Els routers (encaminadors) d'Internet es solen configurar de forma que descarten qualsevol trànsit dirigit a les adreces IP privades. Aquest aïllament proporciona a les adreces privades una forma de seguretat bàsica, ja que no és possible que algú des de fora la xarxa privada estableixi connexió directa a una màquina mitjançant aquestes adreces. Per això no és possible realitzar connexions entre diferents xarxes privades mitjançant Internet, diferents empreses utilitzen el mateix rang d'IP privades sense risc de tenir conflictes, és a dir, no es corre el risc que una comunicació arribi per error a un tercer que estigui utilitzant la mateixa adreça IP privada.
Les adreces privades són:

Nom	Rang d'adreces IP	Nombre d'IP	Descripció de la classe	Major bloc de CIDR (màscara de subxarxa)	Definit a
Bloc de 24 bits	10.0.0.0 – 10.255.255.255	16.777.216	xarxa simple de classe A	10.0.0.0/8 (255.0.0.0)	RFC 1597 (obsolet), RFC 1918
Bloc de 20 bits	172.16.0.0 – 172.31.255.255	1.048.576	16 xarxes classe B contínues	172.16.0.0/12 (255.240.0.0)	
Bloc de 16 bits	192.168.0.0 – 192.168.255.255	65.536	256 xarxes classe C contínues	192.168.0.0/16 (255.255.0.0)	
Bloc de 8 bits	169.254.0.0 – 169.254.255.255	65.536	xarxa simple classe B	169.254.0.0/16	RFC 3330, RFC 3927

Taula 2. Rangs d'adreces privades.

2.1.4. TCP

Transmission Control Protocol (TCP) és un protocol orientat a la connexió dintre del nivell de transport del model OSI que permet l'entrega de paquets de manera fiable, en el cas de TCP anomenats segments. Això significa que abans de poder transmetre cap dada, és necessari establir una connexió entre els dos nodes que es volen comunicar. Un cop establerta la connexió, el protocol s'encarrega de garantir que les dades arriben de manera correcta, ordenada i sense duplicats al punt de destinació. Un cop finalitza la transmissió és necessari tancar la connexió.

Les aplicacions que utilitzen TCP per comunicar-se no s'han de preocupar de la integritat de la informació, no han de fer cap tipus de control d'errors atès que poden assumir que tot el que reben és correcte, el mateix protocol s'encarrega de les tasques de control de flux i d'errors.
La fiabilitat associada a aquest protocol té un cost en la quantitat de recursos necessaris que el fan inadequat per alguns usos, com les retransmissions en temps real de vídeo, en què és preferible descartar un paquet que no ha arribat o que ho ha fet en mal estat atès que per a l'emissor pot resultar difícil retransmetre'l i per al receptor seria un problema esperar. Per a aquests casos existeix el protocol UDP.

2.1.5. UDP

User Datagram Protocol (**UDP**) és un protocol no orientat a connexió del nivell de transport del model OSI, basat en l'intercanvi de datagrames. UDP permet l'enviament de datagrames a través d'una xarxa sense que s'hagi establert prèviament una connexió, ja que el mateix datagrama incorpora suficient informació d'adreçament a la seva capçalera. Tampoc té control ni confirmació del flux, per tant els paquets es poden avançar els uns als altres. Tampoc no se sap si els paquets han arribat correctament, ja que no té cap confirmació d'entrega o recepció. El fet que no s'hagin de reconèixer obligatòriament tots els paquets rebuts (mitjançant ACK, és a dir, un comprovant de recepció), suposa un estalvi de dades (menys overhead, capçalera) que el fa més àgil que altres protocols del mateix nivell de transport orientats a connexió, com és el cas de TCP. És per això que s'acostuma a fer servir quan és més important la rapidesa que la fiabilitat. Per exemple, per transmetre veu o vídeo, aplicacions on resulta més important transmetre amb velocitat que garantir el fet que arribin absolutament tots els bytes. És també útil per a servidors que responen petites consultes d'un gran nombre de clients. A diferència de TCP, és compatible amb la difusió de paquets. S'acostuma a fer servir per protocols DHCP, BOOTP, DNS, TFTP i més, que no explicarem.

2.1.6. TCP/IP

TCP/IP és un conjunt dels dos protocols més importants dels ja explicats, el TCP i l'IP. Aquest conjunt cobreix diversos nivells del model OSI, unint les capes de presentació i de sessió en la capa d'aplicació. Aquests protocols són utilitzats per tots els ordinadors connectats a Internet, de manera que aquests puguin comunicar-se entre si. Cal tenir en compte que a Internet es troben connectats ordinadors de classes molt diferents i amb maquinari i programari incompatibles en molts casos, a més de tots els mitjans i formes possibles de connexió. Aquí es troba un dels grans avantatges del TCP/IP, ja que aquest protocol s'encarregarà que la comunicació entre tots sigui possible. TCP/IP és compatible amb qualsevol sistema operatiu i amb qualsevol tipus de maquinari.

Diagrama 6. TCP/IP

El conjunt TCP/IP està dissenyat per a encaminar i té un grau molt elevat de fiabilitat, és adequat per a xarxes grans i mitjanes, així com en xarxes empresarials. S'utilitza a escala mundial per connectar-se a Internet i als servidors web. És compatible amb les eines estàndard per analitzar el funcionament de la xarxa. S'utilitza tant en campus universitaris com en complexos empresarials, on utilitzen molts encaminadors i connexions a ordinadors UNIX, i també en xarxes petites o domèstiques, en telèfons mòbils i en domòtica.

Ping:

El **ping** és una eina de les xarxes TCP/IP que permet, de forma similar a una trucada perduda, determinar si un servidor està funcionant i si s'hi pot arribar des d'una xarxa concreta amb l'ordinador des d'on es fa la prova. El programa envia paquets ICMP i n'espera les respostes. També és anàleg al funcionament dels sonars en els submarins, amb què es va inspirar el seu autor, *Mike Muuss*.

És una eina molt útil per a la diagnosi de les connexions d'Internet. No obstant això, molts proveïdors de serveis d'Internet filtren o bloquen aquest tipus de sol·licitud.

2.2. Xarxa per Ethernet

Ethernet és una família de tecnologies basades en el marc de les xarxes d'ordinadors per a xarxes d'àrea local (LAN). El nom ve del concepte físic de l'èter. Es defineix una sèrie d'estàndards de cablejat i senyalització de la capa física del model OSI de xarxa, a través dels mitjans d'accés de xarxa amb adr MAC (Media Acces Control)/ Capa d'enllaç de dades i un format d'adreces comú.
Ethernet és un estàndard de xarxa de computadores d'àrea local amb accés al medi per contingut CSMA/CD ("Accés Múltiple per Detecció de Portadora amb Detecció de Col·lisions"), és una tècnica usada en xarxes Ethernet per millorar les seves prestacions.
L'Ethernet es pren com a base per la redacció de l'estàndard internacional IEEE 802.3. Usualment es pren Ethernet i IEEE 802.3 com a sinònims. Es diferencien en un dels camps de la trama de dades. Les trames Ethernet i IEEE 802.3 poden coexistir a la mateixa xarxa. Ethernet va ser estandarditzada com IEEE 802.3 que determina com les màquines de la xarxa envien i reben dades sobre un medi físic compartit que es comporta com un bus lògic, independentment de la seva configuració física. Ethernet va ser desenvolupada als anys 70 al Xerox PARC, per Robert Metcalfe. Actualment Ethernet és l'estàndard més utilitzat en xarxes locals. Des dels anys 1990, es fa servir sovint Ethernet per a les connexions entre clients. Aquesta configuració ha desplaçat altres estàndards com Token Ring, FDDI i ARPANET.

El datalogger estarà pensat per a connectar-lo a través de connexió Ethernet per defecte, tot i que una instal·lació molt llarga de cable por resultar cara.

2.3. Xarxa per USB Wifi

Wifi, sovint escrit com **Wi-fi, WiFi, i WLAN**, és una tecnologia de xarxa local sense fils que permet a un dispositiu electrònic intercanviar dades o connectar amb Internet sigui a 2,4 GHz o 5 GHz. El nom és una marca registrada, i acrònim de *wireless fidelity* ("fidelitat sense cable"). L'aliança Wi-Fi ho descriu com qualsevol "producte wifi de xarxa local basat en l'estàndard 802.11a/b/g/n/ac de la IEEE". Hi ha molts dispositius habilitats per utilitzar comunicació Wi-Fi: ordinadors, impressores, videoconsoles, smatphones, càmeres digitals, tablets, reproductors digitals multimèdia... Aquests es poden connectar a un recurs de xarxa com Internet via un punt d'accés wifi. Aquest punt d'accés wifi té uns 20 metres en interior fins a alguns centenars en camp obert. La cobertura pot ser dins d'una habitació amb parets i obstacles que bloquegen el senyal, fins a diversos kilòmetres utilitzant punts d'accés intermedis amb antenes direccionals.
Wi-Fi pot ser menys segur que les connexions per cable (Ethernet), ja que l'atacant no necessita ser físicament connectat. Les pàgines web que utilitzen SSL són segures però l'accés wifi sense encriptació és fàcilment detectat pels atacants. A causa d'això les xarxes sense fils han desenvolupat tècniques d'encriptació. El primer estàndard d'encriptació, conegut com a WEP, actualment es troba obsolet perquè és extremadament dèbil. Protocols d'encriptació de més alta qualitat,WPA i WPA2 van aparèixer més tard com a solució a WEP. Una funcionalitat especial introduïda el 2007 coneguda com a Wi-Fi Protected Setup (WPS), tenia com a funció permetre connectar dispositius a un punt d'accés amb WPA o WPA-2 prement un botó, WPS té una vulnerabilitat que permet a un atacant obtenir la clau WPA/WPA-2 sense conèixer el pin de configuració WPS, és aconsellable deshabilitar WPS.

Es connectarà a través de connexió USB un dispositiu wifi a la RaspberryPi. El mateix aparell podrà ser configurat com a client DHCP, o com a servidor Access Point.

2.3.1.Mode client DHCP

DHCP (acrònim en anglès de **D**ynamic **H**ost **C**onfiguration **P**rotocol) és un protocol de xarxa que permet als nodes d'una xarxa IP obtenir els seus paràmetres de configuració automàticament. Es tracta d'un protocol de tipus client/servidor en el que generalment un servidor posseeix una llista d'adreces IP dinàmiques i les va assignant als clients a mesura que aquestes van estant lliures, sabent en tot moment qui ha estat en possessió d'aquella IP, quant temps l'ha tinguda o a qui se li ha assignat després.

Sense DHCP, cada adreça IP ha de configurar-se manualment a cada ordinador i, si l'ordinador es mou a un altre lloc en una altra part de la xarxa, s'ha de configurar una altra adreça IP diferent. El DHCP li permet a l'administrador supervisar i distribuir de forma centralitzada les adreces IP necessàries i, automàticament, assignar i enviar una nova IP si l'ordinador és connectat en un lloc diferent de la xarxa.

El protocol DHCP inclou tres mètodes d'assignació d'adreces IP:

-**Assignació manual o estàtica**: Assigna una adreça IP a una màquina determinada. Se sol utilitzar quan es vol controlar l'assignació d'adreça IP a cada client, i evitar, també, que es connectin clients no identificats.

-**Assignació automàtica**: Assigna una adreça IP de forma permanent a una màquina client la primera vegada que fa la sol·licitud al servidor DHCP i fins que el client l'allibera. Se sol utilitzar quan el nombre de clients no varia massa.

-**Assignació dinàmica**: L'únic mètode que permet la reutilització dinàmica de les adreces IP. L'administrador de la xarxa determina un rang d'adreces IP i cada computadora connectada a la xarxa està configurada per sol·licitar la seva adreça IP al servidor quan la targeta d'interfície de xarxa s'inicialitza. El procediment usa un concepte molt simple en un interval de temps controlable. Això facilita la instal·lació de noves màquines clients a la xarxa.

La sessió DHCP es duu a terme de la següent manera;

1r: DHCP Discovery
DHCP Discovery és una sol·licitud DHCP realitzada per un client al servidor DHCP administrador de la xarxa, li assigni una direcció IP altres paràmetres DHCP com la màscara de xarxa o el nom DNS.

2n: DHCP Offer
DHCP Offer és el paquet de resposta del Servidor DHCP a un client DHCP davant la petició de l'asignació dels paràmetres DHCP. Aquí és on s'involucra l'adreça MAC.

3r: DHCP Request
El client selecciona la configuració dels paquets rebuts de DHCP Offer. Un cop més, el client sol·licita una direcció IP específica que indiqui el servidor.

4rt: DHCP Acknowledge
Quan el servidor DHCP rep el missatge DHCPREQUEST del client, s'inicia la fase final del procés de configuració. Aquesta fase implica el reconeixement amb l'envio d'un paquet al client.

Aquest paquet inclou la duració de la concessió i de qualsevol altre informació de configuració que el client pugui tenir sol·licitada. En aquest punt, el procés de configuració TCP/IP s'ha completat. El servidor reconeix la sol·licitud i l'envia cap al client. El sistema en el seu conjunt espera que el client configuri la seva xarxa amb les opcions subministrades. El servidor DHCP respon a la DHCPREQUEST amb un paquet DHCPACK, completant així el cicle de reconeixement. La direcció d'origen és l'adreça IP del servidor DHCP, i la direcció de destí és encara 255.255.255.255. El camp YIADDR conté la direcció del client, i els camps CHADDR i DHCP: Client Identifier són la direcció física de la targeta de xarxa en el client. La secció d'opcions del DHCP identifica el paquet com un ACK.

2.3.2. Mode Wireless Access Point (WAP)

Un **punt d'accés sense fils** (**WAP** o **AP** per les seves sigles en anglès: Wireless Access Point) en xarxes d'ordinadors és un dispositiu que interconnecta dispositius de comunicació sense fils per formar una xarxa sense fils. Normalment un WAP també pot connectar-se a una xarxa cablejada, i pot transmetre dades entre els dispositius connectats a la xarxa cable i els dispositius sense fils. Molts WAPs poden connectar-se entre si per a formar una xarxa encara major, permetent realitzar "roaming". (D'altra banda, una xarxa on els dispositius client s'administren a si mateixos - sense la necessitat d'un punt d'accés - es converteixen en una xarxa ad-hoc. Els punts d'accés sense fils tenen adreces IP assignades, per a poder ser configurats.
Són els encarregats de crear la xarxa, estan sempre a l'espera de nous clients als quals donar serveis. El punt d'accés rep la informació, l'emmagatzema i la transmet entre la WLAN (Wireless LAN) i la LAN cablejada.
Un únic punt d'accés pot suportar un petit grup d'usuaris i pot funcionar en un rang d'almenys trenta metres i fins a diversos centenars. Aquest o la seva antena són normalment col·locats en alt però podria col·locar-se en qualsevol lloc en què s'obtingui la cobertura de ràdio desitjada. L'usuari final accedeix a la xarxa WLAN a través d'adaptadors. Aquests proporcionen una interfície entre el sistema d'operació de xarxa del client (ENS: Network Operating System) i les ones, mitjançant una antena sense fils.

Aquest mode està creat per a poder-se connectar a la pròpia xarxa wifi que genera el dispositiu USB des de un mòbil tipus smartphone o una tablet, que actualment ja es fan servir en l'àmbit industrial.

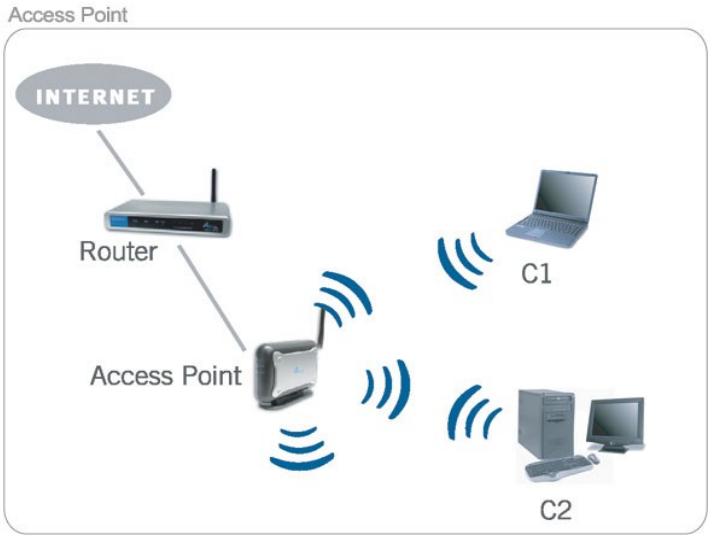


Diagrama 7: Exemple de connexió AccessPoint

2.4. Xarxa per Mòdem 3G

La **Banda Ampla Mòbil** (BAM), també coneguda com ADSM (per ADSL mòbil), és un servei d'Internet Mòbil amb banda ampla.

Aquesta tecnologia permet obtenir Internet en qualsevol lloc i moment, sempre que es disposi de cobertura mòbil, i pot oferir velocitats equiparables a les velocitats de banda ampla per cable (entre 3 i 42 Mbps depenent de l'operador i del tipus de connexió: GPRS, 3G , 4G).

L'International Mobile Telecommunications-2000 (IMT-2000) o altrament dit **3a Generació**(**3G**), són un conjunt d'estàndards i de tecnologies de comunicació mòbil. Es basen en recomanacions fetes per la Unió Internacional de Telecomunicacions (ITU) i el seu objectiu és el de permetre les comunicacions mòbils globals i oferir serveis multimèdia als usuaris de telefonia mòbil.

En el sistema datalogger dissenyat també s'hi podrà connectar un mòdem de de la companyia Vodafone que ofereix xarxa 3G a través de connexió USB.

2.5. Internet per WIMAX

WIMAX ("Worldwide Interoperability for Microwave Access", en català "inter-operabilitat mundial per accés per microones") és una tecnologia que permet realitzar transmissions de dades sense fils i establir connexió a Internet a locatitzacions remotes, inaccessibles per connexions convencionals (cable o fibra òptica). Instal·lant una antena direccional, apuntada a una centraleta de qualsevol companyia que disposi d'aquest servei, garanteix la connexió a internet. Aquesta antena es connecta al router on també es pot connectar un conversor de trucades a paquets IP, la telefonia IP. La veu per IP, també coneguda com VoIP (de l'anglès Voice over IP), és una tecnologia per a mantenir converses amb veu a Internet o a qualsevol xarxa.

Imatge 3. Antena WIMAX

2.6. Xarxa Privada Virtual (VPN)

Una **VPN**, o **xarxa privada virtual(**de les inicials de ***virtual private network)*** és una tecnologia de xarxa que permet una extensió de la xarxa local sobre una xarxa pública o no controlada, com per exemple Internet.

Els avantatges de fer servir una VPN són:

- Integritat, confidencialitat i seguretat de dades.
- Les VPN redueixen els costos i són senzilles d'utilitzar.
- Facilita la comunicació entre dos usuaris en llocs distants.

Per fer-ho possible de manera segura és necessari proporcionar els mitjans per garantir l'autenticació, integritat i confidencialitat de tota la comunicació:

-**Autenticació i autorització**: Qui està de l'altre costat? Usuari/equip i quin nivell d'accés ha de tenir.
-**Integritat**: que les dades enviades no han estat alterats. Per a això s'utilitza *funcions de Hash* . Els algorismes de hash més comuns són els *Message Digest* (MD2 i MD5) i el *Secure Hash Algorithm* (SHA).
-**Confidencialitat**: Atès que només pot ser interpretada pels destinataris de la mateixa, es fa ús d'algorismes de xifrat per encriptar les claus d'accés, com *Data Encryption Standard* (DES), Triple DES (3DES) i *Advanced Encryption Standard* (AES), o com en el cas d'aquest projecte, EasyRSA.
-**No repudi**: és a dir, un missatge ha d'anar signat, i qui el signa no pot negar que el missatge l'ha enviat ell o ella.

Exemples comuns en són la possibilitat de connectar dues o més sucursals d'una empresa utilitzant com a vincle Internet; permetre als membres de l'equip de suport tècnic la connexió des de casa al centre de còmput, i que un usuari pugui accedir al seu equip domèstic des d'un lloc remot, com ara un hotel o una cafeteria. Tot això, utilitzant la infraestructura d'Internet.

Hi ha tres tipus d'arquitectures de connexió VPN:

2.6.1. VPN d'accés remot

És potser el model més utilitzat actualment, i consisteix en usuari sobre proveïdors que es connecten amb l'empresa des de llocs remots (oficines comercials, domicilis, hotels, avions preparats, etc.) utilitzant Internet com a vincle d'accés. Una vegada autenticats tenen un nivell d'accés molt similar al que tenen en la xarxa local de l'empresa.

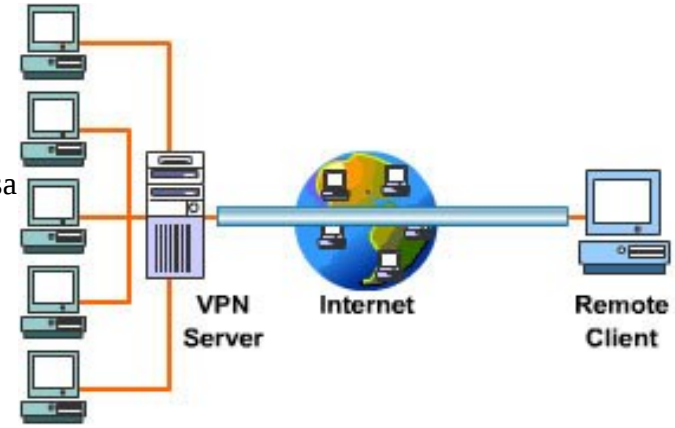


Diagrama 8: Exemple de connexió VPN d'accés remot

2.6.2. VPN punt a punt

Aquest esquema s'utilitza per connectar oficines remotes amb la seu central de l'organització. El servidor VPN, que posseeix un vincle permanent a Internet, accepta les connexions via Internet provinents dels llocs i estableix el túnel VPN. Els servidors de les sucursals es connecten a Internet utilitzant els serveis del seu proveïdor local d'Internet, típicament mitjançant connexions de banda ampla. Això permet eliminar els costosos vincles punt a punt tradicionals (realitzats comunament mitjançant connexions de cable físiques entre els nodes), sobretot en les comunicacions internacionals. És més comú el següent punt, també anomenat tecnologia de túnel o *tunneling* .

Tunneling

La tècnica de tunneling consisteix a encapsular un protocol de xarxa sobre un altre (protocol de xarxa encapsulat) creant un túnel dins d'una xarxa d'ordinadors. L'establiment d'aquest túnel s'implementa incloent una PDU (trama de dades dels nivells superiors a la capa de transport) determinada dins d'una altra PDU amb l'objectiu de transmetre des d'un extrem a l'altre del túnel sense que sigui necessària una interpretació intermèdia de la PDU encapsulada. D'aquesta manera s'encaminen els paquets de dades sobre nodes intermedis que són incapaços de veure en clar el contingut d'aquests paquets. El túnel queda definit pels punts extrems i el protocol de comunicació emprat, que entre altres, podria ser SSH.

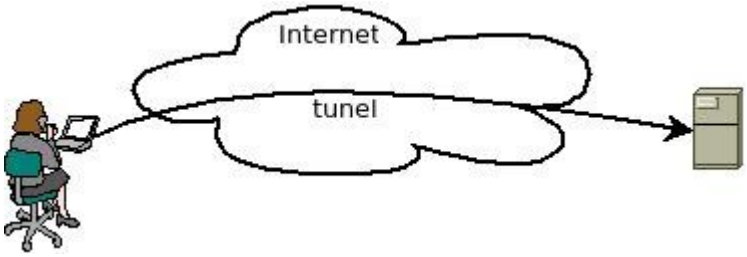


Diagrama 9: Exemple de connexió tunneling

2.6.3. VPN over LAN

Aquest esquema és el menys difós però un dels més poderosos per utilitzar dins de l'empresa. És una variant del tipus "accés remot" però, en comptes d'utilitzar Internet com a mitjà de connexió, utilitza la mateixa xarxa d'àrea local (LAN) de l'empresa. Serveix per aïllar zones i serveis de la xarxa interna. Aquesta capacitat ho fa molt convenient per millorar les prestacions de seguretat de les xarxes sense fils (WiFi).

Un exemple clàssic és un servidor amb informació sensible, com les nòmines de sous, situat darrere d'un equip VPN, el qual proveeix autenticació addicional més l'agregat del xifrat, fent possible que només el personal de recursos humans habilitat pugui accedir a la informació.

Un altre exemple és la connexió a xarxes WIFI fent ús de túnels xifrats IPSEC o SSL que a més de passar pels mètodes d'autenticació tradicionals (WAP, WEP, MACaddress, etc.), agreguen les credencials de seguretat del túnel VPN creat a la LAN internes o externes.

3. HARDWARE DEL SISTEMA

3.1. Raspberry Pi

Raspberry Pi és un ordinador de placa reduïda o placa única (SBC) de baix cost desenvolupat al Regne Unit per la Fundació Raspberry Pi, amb l'objectiu d'estimular l'ensenyament de ciències de la computació a les escoles.
Per al desenvolupament d'aquest projecte s'ha utilitzat la Raspberry Pi Model B+ i les seves característiques són:

- Font d'alimentació dual tipus *buck* de 3,3 V i 1,8 V.
- Alimentació de 600mA fins a 1.8A @ 5V amb protecció de polaritat.
- CPU: Processador Broadcom BCM2835 SoC @ 700MHz.
- GPU: Dual Core VideoCore IV® Multimedia Co-Processor
- 512MB SDRAM @ 400 MHz.
- Connector CSI-2 per a càmera.
- Connector display DSI.
- Port HDMI.
- Targeta de xarxa amb connexió Ethernet 10/100MB per entrada RJ45 Jack.
- 4 ports USB.
- 40 pins GPIO.
- Ranura per a targeta microSD.
- Mides: 85mm x 56mm.
- Fusible de 2A i protecció d'intercanvi en calent.
- Vídeo Compost (NTSC / PAL) integrat en un connector de 3,5 mm de 4 polos per a auriculars. Resolucions des de 640×350 fins a 1920×1200, incloent 1080p.

Imatge 4. Raspberry Pi Model B+

Observacions:

-La font d'alimentació oficial de RaspberryPi és exclusivament per a la placa Raspberry Pi Model B+, no és d'ús general.
-La RaspberryPi utilitzada en aquest projecte té una potència de càlcul baixa per a l'ús que se'n fa en aquest projecte. Per a una major experiència es recomana fer servir un model més nou de la RaspberryPi, amb 1GB de RAM.

3.2. PiFace Digital 2

La placa d'expansió Piface Digital es connecta al port GPIO de la Raspberrypi.

Aquesta placa està dissenyada especialment per incrementar les entrades i sortides de la Raspberry Pi, per als models A + i B +.

PiFace Digital és la interfície més utilitzat per a la Raspberry i que converteix aquesta en un dispositiu capaç d'interactuar amb interruptors, leds, o sensors.

Les característiques són:

- 2 relés (voltatges fins 20V i corrents fins 5A).
- 4 interruptors tàctils.
- 8 entrades digitals.
- 8 sortides de col·lector obert.
- 8 indicadors LED.
- Conté buffers per protegir les E / S de la Raspberry Pi.
- Programable en Scratch o Python

Imatge 5. PiFace Digital 2

3.3. Placa I2C-AI418S 4-20ma DAC

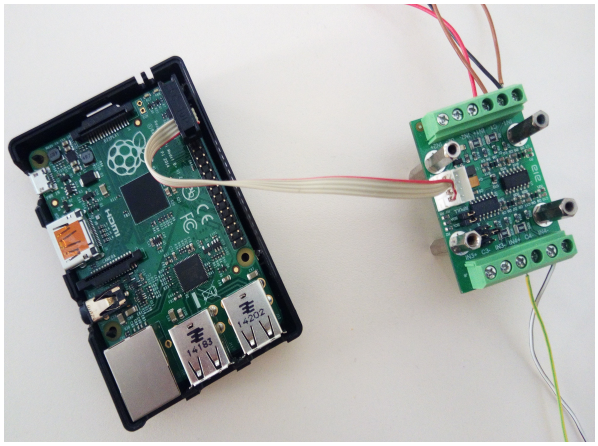
Necessitem una targeta d'entrada analògica per als sensors de bucle de corrent de 4-20 mA estàndard, i em escollit la placa I2C-AI418S ADC. La junta es pot connectar a fonts de tensió i corrent. Disposa de 4 canals. Cada canal accepta de 0-5 volts, 0-10, 0-20, 4-20, 0-40mA. Així que aquesta placa pot connectar-se a un sensors de 4-20 mA. La resolució de bits es pot seleccionar com 12,14,16 i 18 bits. El bit més significatiu (MSB) s'usa per a un bit de signe. El PGA (amplificador) es pot programar com 1,2,4 i 8. Cal el tauler d'interfície al microcontrolador a través del bus I2C que és compatible a 100 Khz, 400 kHz i a busos ràpids de 3.4Mhz de velocitat. La direcció de bus I2C es selecciona mitjançant dos 'jumpers' de la placa. La junta ADC I2C es pot configurar fins a vuit direccions. Significa que vuit taules poden connectar-se entre si en el mateix bus. Així mateix, la junta té resistències de pull-up per bus I2C que poden ser activades o desactivades pels 'jumpers'. L'alimentació de la placa I2C ADC només vol tensió de l'2.7-5.5v.

Llavors, consta de:

- ❒ 4 Channels Of Analog Inputs
- ❒ Up to 32 channels on one bus
- ❒ MCP3424, 12,14,16 and 18-Bit
- ❒ Voltage Input: 0-5V, 0-10V
- ❒ Current Input: 0-20mA, 4-20mA, 0-40mA
- ❒ I2C Bus Interface 100Khz, 400Khz, 3.4Mhz

Imatge 6. I2C-AI418S

- Programmable Gain Amplifier(PGA)
- Pull-Up Resistors for I2C Bus
- Up To 8 Boards On Bus
- Compatible With Most Microcontrollers
- Single Supply Operating Voltage 2.7V to 5.5V
- Inverse Polarity Protection Circuits for Supply Voltage
- Over Input Voltage Protections
- PCB size 37x53mm



Imatge 7. Muntatge I2C-AI418S

* No s'han connectat simultàniament les dues plaques PiFace Digital 2 i I2C-AI418S ja que es requereix d'una soldadura en els busos lliures de la Piface, i no s'ha realitzat.

3.4. Targeta de memòria microSD

La ROM de la Raspberry és extraïble i correspon a un format de targeta de memòria flash desenvolupada per SanDisk; adoptada per l'Associació de Targetes SD (SD Card Association) sota el nom de «microSD» el juliol de 2005.
Fa tan sols 15 × 11 × 0,7 mm, la qual cosa li dóna una àrea de 165 mm².
La tarjeta microSD feta servir en aquest projecte consta de 8GB de capacitat.
Observació: és possible que els problemes de lentitud mencionats a la Raspberry Pi siguin deguts la lentitud de transmissió de dades (lectura i escriptura) entre la Raspberry i la targeta microSD. Per tant es recomana utilitzar una microSD de 16GB i de 10Mb/s de velocitat.

Classe	Velocitat
Classe 2	2 MB/s
Classe 4	4 MB/s
Classe 6	6 MB/s
Classe 10	10 MB/s

Taula 3. Classes i velocitats de les SD

Imatge 8. MicroSD

3.5. Adaptador USB WIFI

Un adaptador Wifi és un aparell amb antena wifi integrada per a poder establir connexió per un port usb, a falta de targetes físiques de xarxa.

En aquest projecte s'ha utilitzat el model Logilink Wireless N 300Mbps USB Adapter 11n ja que les seves característiques eren les adients (suporta SO Linux)

Compleix els estàndards IEEE 802.11b/g/n
Alta velocitat, fins a 300Mbps
Suporta seguretats de tipus 64/128-bit WEP, WPA, WPA2 i WPS.
Suporta Windows 2000/XP/Vista/7/Linux
Abarca de 100 a 300m

Important: l'adaptador usb ha de poder suportar el sistema operatiu Linux.

Imatge 9. Adaptador Wireless

3.6. Mòdem 3G USB

El mòdem 3G USB utilitzat és del model K4505, amb xip Qualcomm MDM8200, fabricat per Huawei en exclusiva per Vodafone. En realitat es tracta d'un Huawei E182E. És el primer mòdem HSPA + de Vodafone. Suporta fins a 21,6 Mbps, però està preparat per MIMO amb 28,8 Mbps mitjançant actualització de firmware.

Imatge 10. Mòdem USB

4. SOFTWARE DEL SISTEMA

4.1. Linux O.S. (Raspbian)

La Fundació Raspberry ofereix un instal·lador de sistemes operatius anomenat noobs, aquest instal·lador proveeix una selecció de diversos S.O. que es descarreguen i instal·len a través d'Internet a més de contenir Raspbian, una de les distribucions recomanades per la fundació.
Entre d'altres sistemes, podem trobar Ubuntu Mate, Snappy Ubuntu Core, Openelec, OSMC, Pidora o Risc OS.
Per al desenvolupament del projecte s'ha utilitzat Debian Jessie (Debian 8.0) que és un sistema operatiu lliure, basat en Raspbian i optimitzat per al maquinari de Raspberry Pi.
Aquest sistema operatiu conté una interfície d'usuari(GUI) simple, però farem servir els terminals de comandes. També podrà ser configurat a través d'una màquina remota via el protocol SSH.

4.2. SSH

SSH (**S**ecure **SH**ell) és un protocol per dur a terme una sessió segura a través d'una xarxa no segura. Es pot fer servir per accedir a màquines remotes a través de la xarxa, gestionar per complet l'ordinador mitjançant l'intèrpret de comandes, i també pot redirigir el trànsit del servidor X per executar programes gràfics si hi ha un servidor X funcionant.
A més de la connexió amb altres màquines, SSH permet copiar dades de forma segura (tant fitxers sols com simular sessions FTP xifrades), gestionar claus de criptografia tipus RSA per no escriure claus al connectar a les màquines i passar les dades de qualsevol altra aplicació per un canal segur de SSH. SSH treballa de forma similar al telnet. La diferència principal és que SSH usa tècniques de xifrat que fan que la informació viatgi per la xarxa de manera il·legible i que cap persona pugui descobrir l'usuari i la contrasenya de la connexió ni el que s'escriu durant la sessió; tot i que és possible atacar aquest tipus de sistema mitjançant atacs REPLAY i manipular d'aquesta manera la informació entre els destins.
Un dels usos més comuns de la comunicació per SSH des de Windows és PuTTY. Linux i Unix ja incorporen clients SSH de forma gairebé nativa.

4.3. Apache2

Apache HTTP Server és un servidor HTTP (de pàgines web) de codi obert multi-plataforma desenvolupat per Apache Software Foundation.
Quan va començar el seu desenvolupament l'any 1995, es basava inicialment en el codi del popular NCSA HTTPd 1.3, però més tard es reescriuria completament.
Apache presenta entre altres característiques missatges d'error altament configurables, bases de dades d'autenticació i negociació de continguts, però va ser criticat per la manca d'una interfície gràfica que ajudi a configurar-lo.
Des d'abril de 1996 Apache ha estat el servidor HTTP més popular a la World Wide Web; des de març de 2006, tot i això, ha experimentat una declinació de la seva quota de mercat, perduda en major part contra Microsoft Internet Information Services i .NET, fet servir per alguns dels grans proveïdors de blogs.
En octubre de 2007 Apache va servir el 27,73% de tots els llocs web, el 30 de novembre de 2007, Apache servia el 50,76% , i cap al 30 d'abril de 2012, Apache servia el 57,56% de tots els llocs web i era un dels servidors més transitats de tots els dominis amb 65,24% d'utilització.

4.4. Hostapd i isc-dhcp-server

Hostapd és un programa per fer de servidor amb punts d'accés i autenticacions. Hostapd implementa una interfície de control que pot ser utilitzada per controlar el servidor hostapd, obtenir informació d'estat i notificar esdeveniments.

Utilitzant el SO de Linux (Host AP, madwifi, Prism54 i alguns drivers utilitzats pel subsistema del nucli mac80211), gestiona l'accés amb la implementació IEEE 802.11 i els autentificadors IEEE 802.1X / WPA / WPA2 / EAP. Està dissenyat per a ser un programa servidor executat de fons (background) i que actua controlant l'autenticació de l'accés a la xarxa.

Isc-dhcp-server és un programa que gestiona automàticament la informació que ha de tenir cada client per poder accedir a la xarxa creada. En configurar una xarxa d'àrea local (LAN), un client ha de tenir certa informació, com l'adreça IP de la seva interfície, l'adreça IP d'almenys un servidor de noms de domini i l'adreça IP d'un servidor en el LAN que serveix com un router a Internet. En la configuració manual que ha d'escriure en aquesta informació per a cada client nou. Amb aquest programa es gestiona el Dynamic Host Configuration Protocol (DHCP) automàticament perquè els clients puguin connectar-se a la xarxa.

Amb la combinació dels programes Hostapd i isc-dhcp-server, farem servir l'adaptador usb per utilitzar-lo com a servidor Access Point.

4.5. Usb-modeswitch i Sakis3g

USB-modeswitch és una eina que s'utilitza per controlar dispositius anomenats USB "flip-flop', amb els quals és possible canviar la seva manera de qualsevol que sigui la seva funció hauria de ser (per exemple, un mòdem), fins i tot si estan en un primer moment reconeixen-lo com un USB dispositiu d'emmagatzematge, com és el cas de la RaspberryPi.

Sakis3g és un script anomenat «all-in-one script» que crea una connexió a Internet a través del mòdems 3G. El programari funciona amb la majoria dels mòdems USB i Bluetooth. Sakis3g s'utilitza amb freqüència en dispositius com ara RaspberryPi and BeagleBone Black.
La seqüència d'ordres es pot utilitzar a través d'una interfície gràfica d'usuari però per a no sobrecarregar més la RaspberryPi, el configurarem mitjançant línies de comandes, tan sols cal indicar el codi (p.e. 12d1:1464) de l'entrada física del mòdem, i habilitar l'entrada a dispositius USB.

Executarem tots dos programes en ordre dins un script per aconseguir establir una connexió 3G automàticament.

4.6. OpenVPN

OpenVPN és un programa creat per James Yonan l'any 2001, que proveeix una connexió segura a partir del programari SSL (Secure Sockets Layer) i VPN. OpenVPN ofereix connectivitat punt- a-punt amb validació jeràrquica d'usuaris i hosts connectats remotament per qualsevol xarxa. Està publicat sota la llicència GPL, de programari lliure. **Transport Layer Security** (TLS) i el seu predecessor **Secure Sockets Layer** (SSL) són protocols que ofereixen comunicacions segures a Internet mitjançant l'encriptació de dades. No entrarem en detalls ja que és molt complexe.

4.7. Pack OMD

El Projecte OMD va ser fundat el 16 de juliol de 2010. Va ser iniciat per Mathias Kettner i fundada per un grup de membres de la comunitat Nagios ben coneguts. Els fundadors, d'esquerra a dreta a l'imatge 11 són: Lars Michelsen (NagVis), Wolfgang Barth (autor de "Nagios System- und Netzwerk-monitoratge."), Gerhard Laußer (Nagios - Das Praxisbuch), Mathias Kettner (Check_MK, Livestatus), Jörg Linge (pnp4nagios), Sven Nierlein (Thruk) i Stefan Hösl.

Imatge 11. Fundadors de OMD

El projecte OMD inclou el programari Nagios, Thruk, pnp4nagios, icinga, check_mk i nagvis entre d'altres. Però només utilitzarem els tres primers.

4.7.1. Nagios

Nagios és un sistema de monitoratge d'equips i serveis de xarxa, escrit en C i sota llicència GNU General Public License versió 2 que permet tenir un complet control de la disponibilitat de serveis, processos i recursos d'equips informant a l'administrador dels problemes abans inclús que els usuaris s'adonin de forma que es pot actuar de forma pro-activa. Inicialment es va anomenar Netsaint, nom que va haver de canviar per coincidència amb una altra marca, va ser creat i actualment mantingut per Ethan Galstad juntament amb un grup de desenvolupadors de programari que mantenen diversos plugins. Nagios va ser dissenyat per a ser executat en Linux tot i que també s'executa en diferents variants de Unix, una d'elles Raspbian, i aprofitarem la ben entesa.

Existeixen dos conceptes pel control i seguiment de les xarxes:
- El host. Aparells físics o nodes de la xarxa.
- El servei. Serveis que ofereixen i pertanyen als hosts.

El sistema va comprovant, en períodes de temps, l'estat dels hosts i els serveis, informació que després és visible per web, utilitzant una altre eina de la fundació més avançada.

Nagios proporciona suport als administradors per a detectar problemes com:
- fallida del servidor de correu
- sobrecàrrega del disc dur
- interrupció de la xarxa

Conté un sofisticat sistema de notificació per informar als administradors quan alguna cosa va malament. És de codi lliure, gratuït, i la fundació posa a disposició de tothom la informació i un apartat de descàrregues on es pot aconseguir altres plugins.

Per a determinar els estats dels hosts o dels serveis, el programa usa quatre estats (OK, WARNING, CRITICAL, UNKNOWN) i es disposa d'un diagrama explicatiu per entendre fàcilment el canvi d'estats.

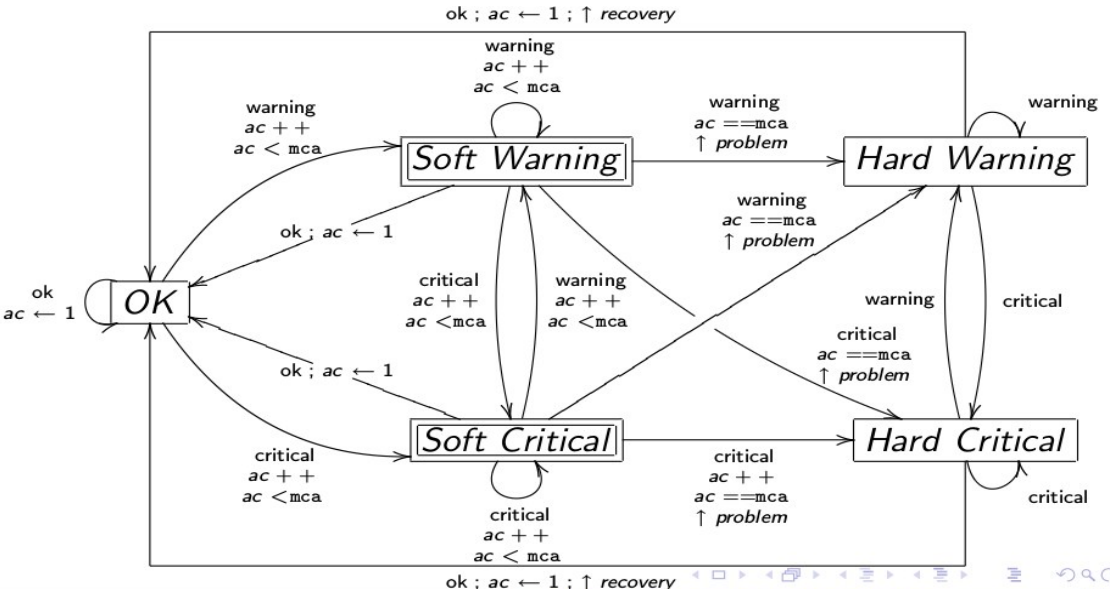


Diagrama 10. Diagrama d'estats del Nagios

On 'ac' és un comptador que s'incrementa a cada comprovació, i 'mca' és el nombre màxim de comprovacions. Existeixen dos tipus de sub-estats (PROBLEM i RECOVERY), que serveixen per a identificar el tipus de notificació. Cada cop que hi ha un canvi d'estat de «soft state» a «hard state», es genera una notificació PROBLEM, quan el canvi d'estat és de «hard state» a «ok», es genera una notificació RECOVERY.

Per a facilitar la programació escriurem tota la nostra configuració amb noves comandes en un únic fitxer. Es mostra com definir hosts, serveis, i les comandes de comprovació.

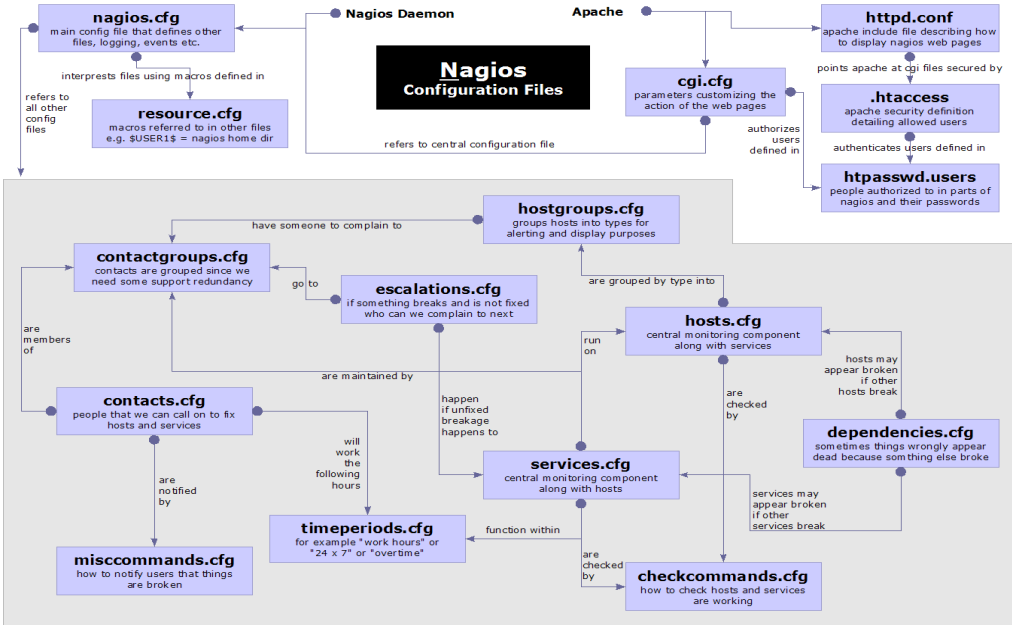


Diagrama 11. Diagrama de fitxers de configuració del Nagios

'hosts.cfg':

```
define host {
        host name               hostexample
        alias                   examplemachine
        address                 152.81.144.22
        check command           check-host-alive
        max check attempts      3
        check period            24 x7
        notification interval   180  # 3 hours
        notification period     24x7
        notification options    d, r, f, u      #down , recovery , flapping ,
                                                #unreachable
        contactgroups           administrators
}
```

'services.cfg':

```
define service {
        host name                       hostexample
        service description             exampleservice

        check command                   check_exampleservice
        max check attempts              3
        normal check interval           5
        retry check interval            1
        check period                    24x7
        notification interval           180
        notification period             24x7
        notification options            w, c, r, f, u # warning, critical, recovery,
                                                      #flapping, unreachable
        contact groups                  administators
}
```

'checkcommands.cfg' :

```
define command{
        command name            check-host-alive
        command line            $USER1$/ check_icmp -H $HOSTADDRESS$
}

define command{
        command name            check_exampleservice
        command line            $USER1$/ check_example -H $HOSTADDRESS$
}
```

'contacts.cfg':

```
define contact {
        contact_name                    fapp
        alias                           administratorfapp
        service_notification_period     24 x7
        host_notification_period        24 x7
        service_notification_options    w, u, c, r
        host_notification_options       d, r
        service_notification_commands   notify-by-email
        host_notification_commands      host-notify-by-email
        email                           fapp
}

define contactgroup {
        contactgroup_name       administrators
        alias                   group_of_administrators
        members                 fapp
}
```

Cada cop que es comprova un host o un servei, s'invoca la seva comanda de comprovació (check_command) que crida a un plugin creat pel nagios o fet per l'administrador.

Per realitzar una notificació de serveis en correu electrònic, s'invoca al plugin notify-by-email :

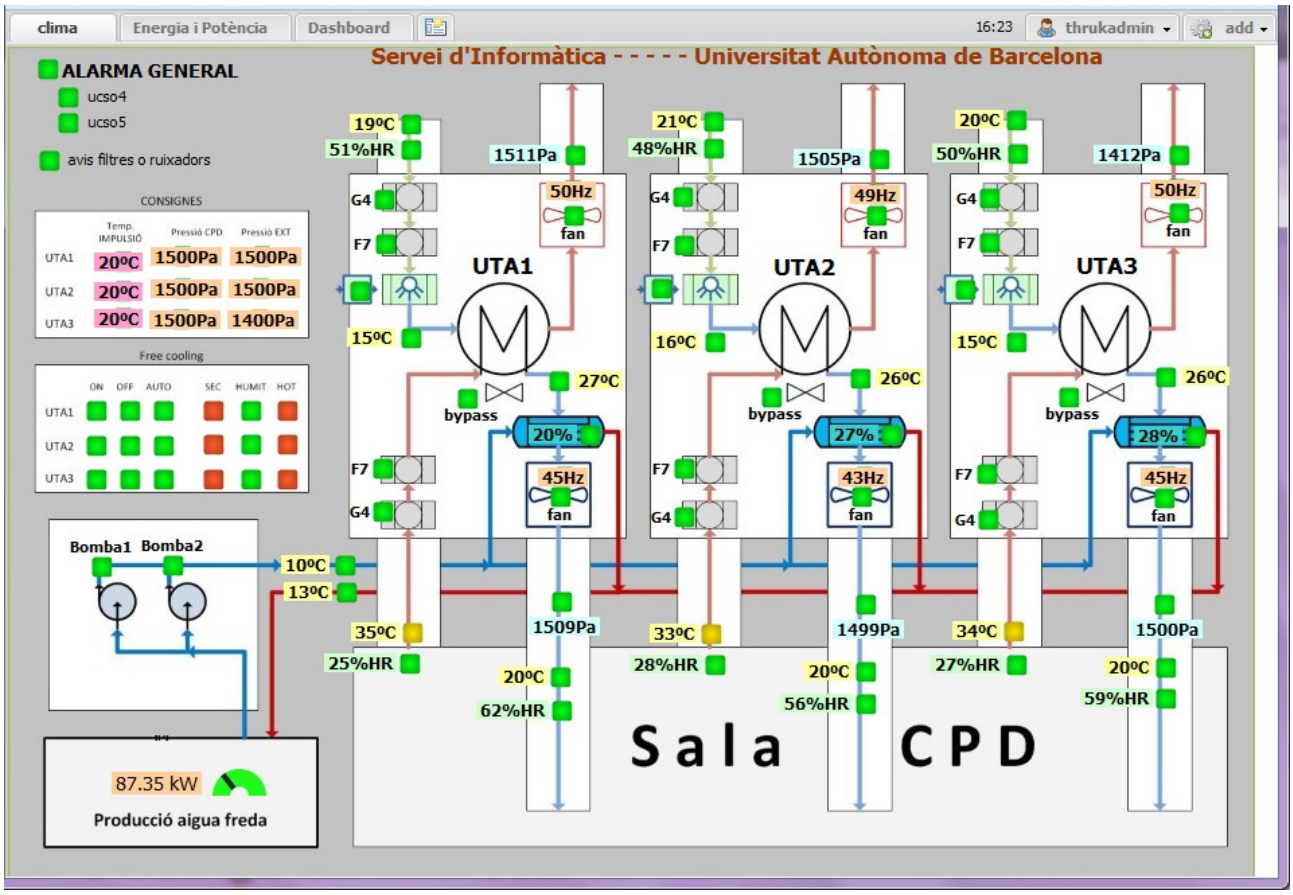
```
service_notification_commands           notify-by-email
```

Crearem la comanda notify-by-led a l'apartat de configuració del Nagios.

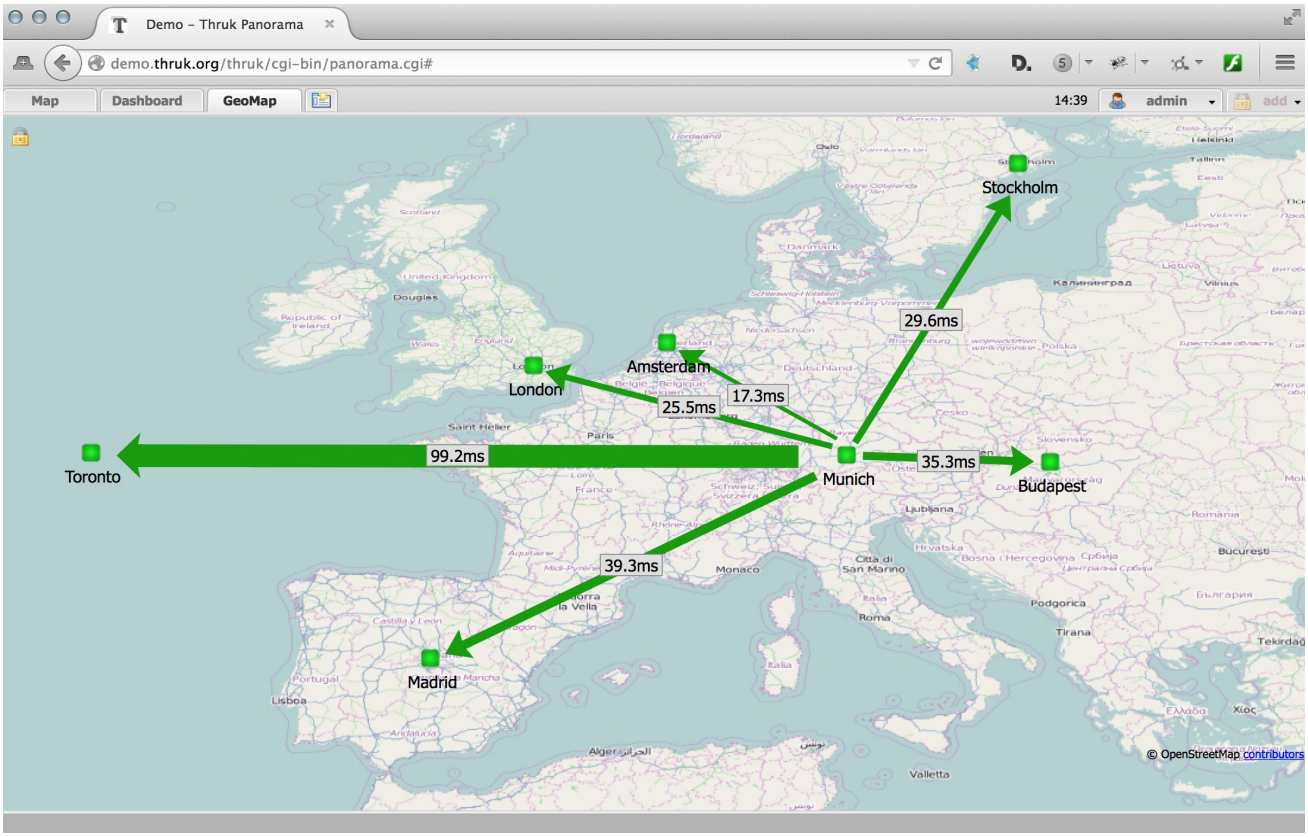
En un navegador, no utilitzarem la interfície gràfica del Nagios, ni del seu successor Icinga, ja que ens interessen les característiques de la nova interfície gràfica proporcionada per a la fundació, el Thruk.

4.7.2. Thruk

El Thruk és la part més nova del programari que s'encarrega de gestionar la interfície gràfica (GUI) del Nagios, per a poder ser visualitzat en un navegador d'Internet.
Thruk és una represa completa de la interfície clàssica en Perl. Mentre es manté l'aspecte original, aporta una gran quantitat de millores i noves característiques. De la mateixa manera, dóna suport a la visualització de múltiples programes, com les gràfiques del Pnp4nagios4 entre d'altres. Thruk incorpora una versió d'interfície gràfica reduïda per a ser mostrada en un mòbil tipus smartphone. També conté una eina anomenada 'Panorama View' que conté Dashboards (taulers), on es pot crear una interfície gràfica personalitzada per al seguiment dels serveis i els hosts. Veiem algun exemple.



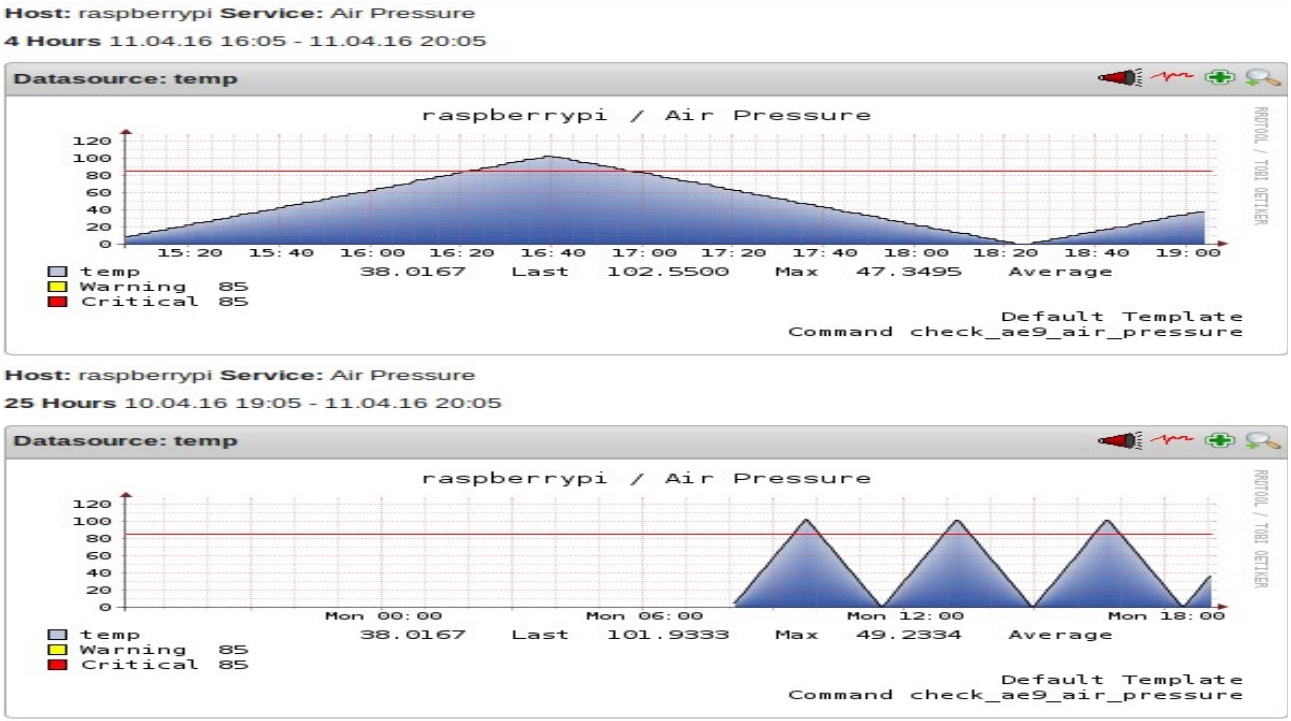
Captura 1. Exemple de monitoratge de sensors de temperatura i pressió sistema de calefacció



Captura 2. Exemple de monitoratge de temps de resposta d'aparells en diferents països

4.7.3. Pnp4nagios

Pnp4nagios és un programa dintre del pack omd i complementa el Nagios analitzant les dades de rendiment proporcionades pels plugins i els emmagatzema automàticament en bases de dades RRD (Round Robin Databases), per a mostrar les dades en gràfiques.



Captura 3. Gràfica del sensor de pressió d'aire simulat. Pnp4nagios.

-Per als fitxers de configuració del **Nagios** i de **OpenVPN** s'ha utilitzat aquest color de fons.

-Per als **fitxers** que calen modificar, s'utilitza aquest color de fons.

-Les línies de **comandes** estan amb aquest color de fons.

-Els **scripts** utilitzats es mostren amb aquest color de fons.

5. Programació

5.1. Programació en C i Python

-El llenguatge de programació C va ser creat per Dennis Ritchie i Ken Thompson als Laboratoris Bell d'AT&T, a principis de la dècada dels 70. C està basat en un llenguatge que havia creat Ken Thompson anomenat llenguatge B el 1970. El llenguatge C es va crear per la necessitat de tenir-ne un que fos més flexible que l'assemblador a l'hora de programar, però que mantingués la característica de ser un llenguatge proper a la màquina. En la seva absència, el llenguatge C es va fer per a poder crear el sistema operatiu Unix. Actualment, C és el llenguatge més utilitzat per a desenvolupar sistemes operatius i altres tipus de programari bàsic, i també per aplicacions en general.
Per a la conversió de els senyals de voltatge o de corrent de la placa I^2C, és necessari crear la connexió 'master-slave' amb la Raspberry, per això ja existeixen unes llibreries i exemples creats que es descarreguen d'Internet. Utilitzarem la biblioteca BCM-2835 versió 1.42, que s'utilitza per a la comunicació amb el xip intern MCP3422/3/4.

El registre de configuració per defecte és

R/W-1	R/W-0	R/W-0	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0
RDY	**C1**	**C0**	**O/C**	**S1**	**S0**	**G1**	**G0**
1	**0**	**0**	**1**	**0**	**0**	**0**	**0**

On:

bit 7 RDY: Ready Bit
One-Shot Conversion mode:
1 = Initiate a new conversion
0 = No effect

bit 6-5 C1-C0: Channel Selection Bits
00 = Select Channel 1 (Default)
01 = Select Channel 2
10 = Select Channel 3 (MCP3424 only, treated as “00” by the MCP3422/MCP3423)
11 = Select Channel 4

bit 4 O/C: Conversion Mode Bit
1 = Continuous Conversion Mode (Default). The device performs data conversions continuously
0 = One-Shot Conversion Mode. The device performs a single conversion and enters a low power standby mode until it receives another write or read command

bit 3-2 S1-S0: Sample Rate Selection Bit
00 = 240 SPS (12 bits) (Default)
01 = 60 SPS (14 bits)
10 = 15 SPS (16 bits)
11 = 3.75 SPS (18 bits)

bit 1-0 G1-G0: PGA Gain Selection Bits
00 = x1 (Default)
01 = x2
10 = x4
11 = x8

Per tant crearem un programa en C a l'apartat 7.6.2 que configuri la placa I²C per a llegir valors de corrent, tan sols canviarem el mode de conversió per a que realitzi una lectura per cada petició de comprovació del Nagios. Tampoc cal més velocitat o resolució que la per defecte, per tant finalment ens quedarà aquesta 'paraula' de registre de configuració:

RDY	C1 *	C0 *	O/C	S1	S0	G1	G0
1	0	0	0	0	0	0	0

*Canviant els camps de (C1 i C0) corresponent als 4 canals.

-**Python** és un llenguatge de programació d'alt nivell i propòsit general molt utilitzat. Va ser creat per Guido van Rossum l'any 1991. La seva filosofia de disseny busca llegibilitat en el codi i la seva sintaxi permet als programadors expressar conceptes en menys línies de codi del que seria possible en llenguatges com C. També proveeix estructures per permetre programes més entenedors tant a petita com a gran escala.
Python suporta diversos paradigmes de programació, incloent-hi programació orientada a objectes, imperativa i també funcional o procedimental. Presenta un sistema dinàmic i una gestió de la memòria automàtica i té una gran i exhaustiva biblioteca estàndard.
Com altres llenguatges de programació dinàmics, Python és usat sovint com a un llenguatge script, però també es fa servir en una àmplia gamma de contextos no-script. Utilitzant eines desenvolupades per tercers, el codi Python pot ser reduït a programes executables independents. Existeixen intèrprets de Python per molts sistemes operatius diferents.
Utilitzarem un programa en Python que permeti gestionar l'encesa de leds envers les notificacions que generi el Nagios.

5.2. Programació Shellscripting

Una Shell d'Unix, és un intèrpret d'ordres, el qual consisteix en la interfície d'usuari tradicional dels sistemes operatius basats en Unix (Linux, Ubuntu, Raspbian...).
Mitjançant les ordres de Shell, l'usuari pot comunicar-se amb el nucli i per extensió, executar aquestes ordres, així com eines que li permeten controlar el funcionament del dispositiu.
Aquestes ordres, poden executar-se ordenadament si s'escriuen en fitxers executables denominats Shell-scripts, d'aquesta manera, quan l'usuari necessita fer ús de diverses ordres o combinats d'ordres amb eines, escriu en un fitxer de text marcat com executable, les operacions que posteriorment, línia per línia, l'intèrpret traduirà al nucli perquè les realitzi. Sense ser estrictament un llenguatge de programació, al procés de crear scripts de Shell se li denomina programació Shell (Shell scripting).
Els usuaris de Unix, poden triar entre diferents Shell (programa que s'hauria d'executar quan inicien la sessió, vegeu bash, ash, csh, zsh, ksh, tcsh). Les interfícies d'usuari per a Unix, poden ser gràfics o de text simple, com són GNOME, KDE i Xfce que poden ser cridades per Shell visuals o Shell gràfiques. (Els usuaris poden especificar quin intèrpret d'ordres desitgen com Shell).

L'estàndard en Unix va acabar sent Bourne shell, ".sh". El programa es troba dins de la jerarquia de fitxers de Unix a "/ bin / sh". En alguns sistemes, "/ bin / sh" és un enllaç simbòlic a un Shell compatible amb més característiques (com Bash).
Els intèrprets d'ordres estan dissenyats per facilitar la forma en què s'invoquen o executen els diferents programes disponibles al computador.

Un script o arxiu d'ordres, es un arxiu amb línies de comandes de processos que s'executen en ordre. Bàsicament és un programa, usualment simple, que s'emmagatzema en un arxiu de text. L'ús habitual dels scripts és realitzar diferents tasques com interactuar amb el sistema operatiu o amb l'usuari.

Per a Linux, els fitxers script solen ser identificats pel sistema a través d'un dels següents encapçalaments en el contingut de l'arxiu, conegut com shebang:

```
#!/bin/bash; #!/bin/ksh; #!/bin/csh; #!/bin/sh
```

També poden ser identificats a través de l'extensió ".sh", sent aquesta potser menys important que l'encapçalament, ja que gairebé tots els sistemes no necessiten aquesta extensió per executar el script, per tant, aquesta sol ser afegida per tradició, o més aviat, és útil perquè l'usuari pugui identificar aquests arxius a través d'una interfície de línia d'ordres sense necessitat d'obrir-lo.
Els scripts són més aviat un programa que li dóna instruccions a altres més avançades, en el nostre cas serviran per a realitzar les configuracions automàticament i per crear simulacions.

-El primer script de configuració realitzat s'ha creat amb la intenció d'agilitzar la connexió a Internet cada cop que em desplaçava de casa a la universitat, l'hem anomenat choosenet.sh i bàsicament modifica el fitxer /etc/network/interfaces i executa les ordres necessàries per a dur a terme la configuració automàticament.

Aquest és el fitxer interfaces.casa:

```
 interfaces(5) file used by ifup(8) and ifdown(8)

# Please note that this file is written to be used with dhcpcd
# For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'
# Include files from /etc/network/interfaces.d:
source-directory /etc/network/interfaces.d

auto lo
iface lo inet loopback

iface eth0 inet static

auto wlan0
allow-hotplug wlan0

# Configuració client dhcp amb wpa a casa
iface wlan0 inet static
  wpa-ssid    WLAN_484
```

```
        wpa-psk     "NzUaRaWFRD"
```

Aquest es el fitxer interfaces.eet:

```
        interfaces(5) file used by ifup(8) and ifdown(8)

        # Please note that this file is written to be used with dhcpcd
        # For static IP, consult /etc/dhcpcd.conf and 'man dhcpcd.conf'
        # Include files from /etc/network/interfaces.d:
        source-directory /etc/network/interfaces.d

        auto lo
        iface lo inet loopback

        iface eth0 inet static

        auto wlan0
        allow-hotplug wlan0

        # Configuració client dhcp amb wpa a UPC
        iface wlan0 inet dhcp
         wpa-ssid    Telematics
         wpa-psk     "xxxxxxxx"
```

Llavors, l'script choosenet.sh:

```
        echo 1- casa
        echo 2- eet
        echo -n "Select option: "
        read val

        if [ $val -eq 1 ]; then
           ln -sf /etc/network/interfaces.casa /etc/network/interfaces
           ifdown wlan0 2> /dev/null >&2
           ifup wlan0
        elif [ $val -eq 2 ]; then
          ln -sf /etc/network/interfaces.eet /etc/network/interfaces
          ifdown wlan0 2> /dev/null >&2
          ifup wlan0
           route add default gw 192.168.10.166
           echo "nameserver 147.83.2.3" > /etc/resolv.conf
        else
           echo "Error: $val option not recognized"
           exit 1
        fi
        exit 0
```

-Un altre script utilitzat es el vodafone.sh que crida als programes usb-modeswitch i sakis3g seguidament per a establir connexió a Internet via 3G.

```
        if [ "$1" == start ]; then
         # Check if usb is connected
          lsusb | grep -E "12d1:(1464|1521)" > /dev/null
         if [ $? -ne 0 ]; then
                echo "Error: Modem not present"
                exit 1
          fi

  usb_modeswitch -W -I -v 0x12d1 -p 0x1521 -M
55534243123456780000000000000011060000000000000000000000000000

        sakis3g connect --console  --debug MODEM="12d1:1464"
        USBINTERFACE=0 SIM_PIN=2444 APN="airtelnet.es"
        APN_USER="vodafone" APN_PASS="vodafone" DIAL="*99#"

elif [ "$1" == stop ]; then
        sakis3g disconnect --console  MODEM="12d1:1464" USBINTERFACE=0
        SIM_PIN=2444 APN="airtelnet.es" APN_USER="vodafone"
        APN_PASS="vodafone" DIAL="*99#"
else
   echo "$0 (start | stop)"
fi
```

-Un cop finalitzat el projecte, el sistema data-logger podrà ser configurat automàticament mitjançant un script de configuració que interactuarà amb l'usuari i preguntarà com es desitja realitzar la configuració de les xarxes de la Raspberry Pi. Per defecte està configurat com a client DHCP a través d'Ethernet. S'ha reciclat l'script 'choosenet.sh' i s'ha situat a /usr/local/bin per a ser executat des de qualsevol directori:

```
        #Funció per parar el servei hostapd en cas de que estigui obert
        function stop_hostapd() {

          service hostapd status > /dev/null 2>&1
          if [ $? -eq 0 ]; then
              service hostapd stop > /dev/null 2>&1
         fi
        }

        echo  "Select option: "
        echo 1- Ethernet
        echo 2- Wifi
        echo 3- Wifi Access Point
        echo 4- 3G
```

```
read val
        if [ $val -eq 1 ]; then
          stop_hostapd
          ifdown eth0 2> /dev/null
          ifdown wlan0 2> /dev/null
          ln -sf /etc/network/interfaces.eth0 /etc/network/interfaces
          ifup eth0
        elif [ $val -eq 2 ]; then
          stop_hostapd
          ifdown eth0 2> /dev/null
          ifdown wlan0 2> /dev/null
          ln -sf /etc/network/interfaces.wlan0 /etc/network/interfaces
          ifup wlan0
        elif [ $val -eq 3 ]; then
          ifdown eth0 2> /dev/null
          ifdown wlan0 2> /dev/null
          ln -sf /etc/network/interfaces.wlan0AP /etc/network/interfaces
          ifup wlan0
          service hostapd start
          /etc/init.d/isc-dhcp-server start
        elif [ $val -eq 4 ]; then
          /home/pi/vodafone.sh
        else
          echo "Error: $val option not recognized"
          exit 1
        fi
        exit 0

# Configuració interfaces.eth0
#       iface eth0 inet manual
#               address 172.16.88.136
#               netmask 255.255.255.0
#               network 172.16.88.0
#               broadcast 172.16.88.255
#               gateway 172.16.88.1

# Configuració interfaces.wlan0
#       allow-hotplug wlan0
#       iface wlan0 inet dhcp
#               wpa-ssid    WLAN_XXX
#               wpa-psk     "xxxxxxxx"

# Configuració interfaces.wlan0AP
#       allow-hotplug wlan0
#       iface wlan0 inet dhcp
#               address 192.168.42.1
#               netmask 255.255.255.0
```

```
#               network 192.168.42.0
#               broadcast 192.168.42.255
```

5.3. Programació dels plugins del Nagios

Un **connector** (en anglès ***plugin***, de *plug-in*: "endollar"), també conegut com a **extensió** (en anglès *addin*, *add-in*, *addon* o *add-on*) és una aplicació informàtica que interacciona amb una altra aplicació (en el nostre cas el Nagios) per aportar-li una funció o utilitat addicional, generalment molt específica, com per exemple servir com a controlador en una aplicació, per a fer així funcionar un dispositiu en un altre programa. Aquesta aplicació addicional és executada per part de l'aplicació principal. Els típics plugins tenen la funció de reproduir determinats formats de gràfics, dades multimèdia, codificar/descodificar correus electrònics, filtrar imatges de programes gràfics, etc.

Cada cop que el Nagios realitza una comprovació, executa un plugin que realitza els processos necessaris per a obtenir els valor de retorn, que s'interpreten amb aquest format:

```
echo -n "Missatge"

echo "| Valors=Valor;valorWARNING;valorCRITICAL;ValorMin;ValorMax"
```

Per exemple:

```
echo -n "Cabal d'aire = $y m3/h "

echo "| Airflow=$y;4700;4500;0;5000"
```

Nagios conté molts exemples de plugins situats a /omd/sites/fapp/lib/nagios/plugins i codificats en binari. S'han utilitzat els de comprovació dels serveis SSH, HTTP i utilització del disc dur. Tots els altres els s'han programat exclusivament per aquest projecte.

En comptes de programar plugins, crearem scripts de bash.

-El primer exemple d'script s'ha usat al Nagios per a comprovar la temperatura de la Raspberry Pi:

```
#! /bin/bash
temp=$(($(</sys/class/thermal/thermal_zone0/temp) / 1000))
freq=$(($(</sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq) /
1000))
echo -n "Raspberrypi temperature is $temp deg.(C) "
echo "| temp=$temp;85;85;0;100 cpufreq=$freq;1200;1200;0;1200"
```

Per a la simulació de la resta de sensors (cabal d'aire, potència, pressió, i capacitat) em utilitzat un senzill script que representa una funció anomenada dents de serra (veure Captura 3 de l'apartat 4.7.3. Pnp4nagios).
Cada cop que el Nagios crida l'script, aquest incrementa o decreix una variable en funció de la pendent d'una recta, que està acotada per valors superior i inferior.

S'han de guardar les variables últim valor i pendent a la memòria del sistema.

-Exemple de simulació de cabal d'aire (airflow):

```
#! /bin/bash

y_max=5000 (valor màxim de la dent de serra)
y_min=4400 (valor mínim de la dent de serra)

y_warn=4700 (valor en estat WARNING)
y_crit=4500 (valor en estat CRITICAL)

#Estats:
RV=0
MSG[0]="OK"
MSG[1]="WARNING"
MSG[2]="CRITICAL"

#Crear variables de 'últim valor' i 'pendent' i modificar-les
if [ -f /tmp/last_y2 ]; then
  read last_y < /tmp/last_y
  read pend < /tmp/pend
else
 last_y=4991
 pend=10
fi
if [ $last_y -gt $y_max ]; then
   echo -10 > /tmp/pend
   pend=-10
elif [ $last_y -le $y_min ]; then
  echo 10 > /tmp/pend
  pend=10
fi

y=$((last_y + pend))
echo $y > /tmp/last_y2

if  [ $y -le $y_crit ]; then
   RV=2
elif [ $y -le  $y_warn ]; then
   RV=1
fi
#Sortida interpretada per Nagios
echo -n "I2C airflow sensor is $y m3/h: ${MSG[$RV]} "

echo "| Airflow=$y;4700;4500;0;5000"

exit $RV
```

Finalment, l'últim script és orientatiu ja que només s'ha provat amb fonts d'alimentació variables, i no amb sensor reals. Quan l'script és executat pel Nagios al realitzar una comprovació, s'invoca al programa en C per a llegir la dada dels sensor:

- On $factor és el factor de conversió inventat per passar les dades de corrent en 'mA' a 'bars'.

```
#! /bin/bash

#Es crida al programa en c per obtenir el valor del sensor

diri2c="/home/pi/bcm2835/bcm2835-1.42/examples/i2c"

factor=35

valor= tail ($diri2c/get_AI481Sv2 -p 1 -r 12 -g 1 -c 0x68)
# La sortida és "Code: 0xba (186d),  Current (mA): 4.0582"

pres=$valor*$factor

if  [ $pres -gt $crit ]; then
  RV=2
elif [ $pres -gt  $warn ]; then
 RV=1
else
  RV=0
fi

warn=625
crit=650
min=0
max=700

MSG[0]="OK"
MSG[1]="WARNING"
MSG[2]="CRITICAL"

echo -n "The pressure is $pres bars: ${MSG[$RV]} "
echo "| Pressure=$pres;$warn;$crit;$min;$max"

exit $RV
```

6. PRESSUPOST

6.1. Pressupost genèric

Com que tot el software utilitzat és lliure i gratuït, només cal comprar el hardware:

Producte	Imatge	Preu
Raspberry Pi Model B_		44.95 €
PI FACE Digital 2		27.24 €
MicroSD SanDisk Ultra 8GB		5,99 €
Raspberry Pi and PiFace Case *Opcional, no s'ha utilitzat en el projecte		7.99 €
I2C-AI418S 4-20ma DAC		25.76 €
	SubTotal:	**111,93 €**

Connectors de 4 pins de 2mm i Cable de 12",4W, 2mm 4F/4F A2804A12F4F4A	 	~3 €
	Total generic:	**~115 €**

El temps que requereix el sistema per a que sigui configurat i muntat es pot comptar depenent dels requeriments del muntatge, però, aproximadament es comptabilitza que en una setmana pot estar el sistema configurat per un professional, cobrant com a mínim 500€:

12,5€/h*8hores*5dies= 500€

6.2. Pressupost adaptador USB Wifi

-Per a realitzar el montatge amb l'adaptador USB:

Logilink N150 adaptador WiFi *No cal si es vol utilitzar per Ethernet o per mòdem 3G		25,99 €
	Total:	**~137€**

6.3. Pressupost Ethernet

-Si es requereix instal·lació de cable RJ-45 per a connexió Ethernet, aproximadament es pot realitzar el muntatge en menys d'un mes, si el projecte no necessita gaires canvis per a ser implementat finalment en un captador de pols industrial.

Cable RJ-45 Ethernet		0,25 €/m
100 metres	**SubTotal:**	**25 € / 100 m**

6.4. Pressupost mòdem 3G USB

-Per realitzar el montatge amb el mòdem 3G USB és compta que es realitzarà una utilització freqüent d'el modem 3G USB de la companyia Vodafone, i pot generar un tràfic elevat, per tant és recomanable contractar una tarifa que s'escaigui a les necessitats. Per un preu de 39 € amb un compromís de permanència de 18 mesos, es garanteix la connexió a Internet, però existeix la possibilitat d'utilitzar la targeta SIM interna per a recarregar el saldo, i s'ha carregat 5€ al mes amb tràfic limitat per a realitzar proves. Vodafone otorga una IP pública que és 169.254.12.135.

Mòdem 3G USB Huaway K4505 *No cal si es vol utilitzar per Ethernet, AP o per wifi.		39 €/mes
	Total:	€ amb 39 €/mes

Les URLS consultades estan a la bibliografia.

7. DESENVOLUPAMENT

En aquest apartat s'explica el procediment per a configurar manualment el nostre sistema i tindrà una estructura de tutorial, per a que qualsevol pugui realitzar el muntatge del datalogger.
A part del material posat al pressupost, per a realitzar la configuració de la Raspberry, es necessitarà:

-Connexió a Internet amb un router per a poder descarregar el sistema operatiu que volem instal·lar a la Raspberrypi
-Ordinador per a poder instal·lar el SO a la targeta de memòria microSD. Ha de disposar de dispositiu d'entrada de de targetes SD/microSD o disposar d'un USB amb ranura per a targetes microSD. A ser possible amb sistema operatiu Linux per a facilitar la connexió SSH.
-Teclat, ratolí, i monitor HDMI per a poder configurar la Raspberry Pi manualment.

7.1. Procediment d'instal·lació del SO a la Raspberrypi

Accedir a https://www.raspberrypi.org/downloads/raspbian/ per a descarregar el fitxer .img de raspbian Jessie.

7.1.1. Instal·lació de Raspbian a la targeta de memòria microSD

Abans de fer l'instal·lació de Raspbian a la targeta de memòria microSD, ens em d'assegurar que l'ordinador fet servir reconeix la targeta de memòria microSD, si no és així canvia de dispositiu d'entrada de l'ordinador per exemple amb un USB per a targeta de memòria microSD.
Cada sistema operatiu té una manera diferent d'instal·lar imatges en perifèrics. En els següents apartats s'explica com fer-ho per a cada un d'ells.

7.1.1.1. Instal·lació des de Linux

Per a que es pugui instal·lar des de Linux el SO descarregat en una targeta de memòria, s'utilitzarà la línia d'ordres del sistema. Per al desenvolupament d'aquest projecte es recomana seguir aquests passos;

a) Entrar al sistema com a 'root' (administrador):

```
$sudo -s
```

i posar contrasenya.

b) Situar-nos en la carpeta on hem descarregat l'arxiu cal tenir en compte l'idioma de l'ordinador fet servir, en aquest cas està en castellà i es diu 'Descargas'.

```
$cd Descargas
```

Es descomprimeix l'arxiu zip per obtenir la imatge.

```
$unzip 2015-11-21-raspbian-jessie.zip
```

c) El darrer pas és desmuntar la targeta microSD. La targeta tindrà un nom similar a '/dev/sdb1'.

```
$umount /dev/sdb1
```

d) En l'últim pas, cal escriure la imatge en la SD amb la següent comanda:

```
$dd bs=1M if=2015-11-21-raspbian-jessie.img of=/dev/sdb
```

7.1.1.2. Instal·lació des de Windows

Per fer la instal·lació des de Windows, hi ha diversos programes molt útils i gratuïts. Per al desenvolupament d'aquest projecte s'ha fet servir el ImageWriter, ja que no funcionava el Fedora, un dels recomanats.

Descarregar ImageWriter des de Softonic.

No cal instal·lació, al fer dos clics, s'obre el programa i s'especifica el nom de l'arxiu zip i el nom de la targeta micro SD (p.e. F: o G:).

És recomanable extreure qualsevol altre memòria externa per facilitar l'identificació de la targeta de memòria.

7.1.1.3. Instal·lació des de Mac

Perquè des MAC es pugui instal·lar el SO descarregat en una targeta de memòria , hi ha diversos programes i scripts molt útils i gratuïts. Per al desenvolupament d'aquest projecte es recomana utilitzar el script de Ray Vijoen , que fa realment fàcil la creació de la targeta de memòria. Es tracta d'un Shellscript que recorre els passos necessaris, inclòs el format, i només es requereix executar una línia d'ordres. Per a això seguirem aquests passos;

a) Descargar Raspberry-Pi-SD-Installer-OS-X des de:
https://github.com/RayViljoen/Raspberry-PI-SD-Installer-OS-X

b) Descomprimir l' arxiu ".zip" i guardar la imatge del SO a la carpeta resultant. L'script admet tenir més d'un arxiu imatge, amb diferents SO. Es pot utilitzar Finder per moure l'arxiu d'imatge a la carpeta.

c) Obrir un terminal d'ordres per accedir a la carpeta i començar a executar l'script.

```
$cd /ruta/fins/directori/Raspberry-Pi-SOTA-Installer-OS-X-master'
```

d) És recomanable extreure qualsevol altre memòria externa per facilitar la identificació de la targeta de memòria. Per comprovar els discos de memòria utilitzar la comanda:

```
$df – hl
```

e) Executar l'instal·lador amb la següent comanda de terminal ;

```
sudo .(install 2015-02-16-wheezy-raspbian.img)
```

f) En quan confirmem l'execució amb la contrasenya del MAC, a la terminal es visualitzaran els dispositius de memòria reconeguts.

g) Introduir el número que pertanyi a la targeta de memòria microSD i esperar que finalitzi la instal·lació de la imatge.

* Assegurar que la unitat de memòria [Filesystem] de destinació és realment la targeta de memòria microSD. Tot el que hi hagi a la microSD quedarà formatada per instal·lar el nou SO. El nom del dispositiu hauria de ser similar a '/Dev/disk2s1'
* Es pot comprovar el progrés de la instal·lació prement Ctrl-T
h) Quan la instal·lació hagi finalitzat , la pantalla del terminal mostrarà la confirmació "All Done!" I la targeta de memòria estarà a punt per funcionar.

7.2. Configuració de Raspbian

Un cop instal·lat el sistema operatiu, es procedeix a la seva configuració. En aquest punt hem d'inserir la targeta de memòria a la ranura corresponent, amb la Raspberry Pi apagada, a més de tenir correctament connectada la nostra Raspberry Pi amb els seus corresponents perifèrics; teclat, ratolí, i monitor HDMI.
En connectar al corrent a la nostra Raspberry, el nou sistema operatiu instal·lat a la targeta de memòria començarà a carregar-se i podrem iniciar la seva configuració.

Abans de tot, entrar al sistema operatiu Raspbian com a 'root' (administrador):

```
pi@raspberrypi:~ $sudo -s
```

No demana posar contrasenya, per defecte és usuari 'pi', contrasenya 'raspberry'.

Abans de començar la configuració, recordem comandes útils, ja que s'utilitzarà l'intèrpret de comandes:

Canviar directori:
```
$cd /ruta/destí/
```
Copiar archiu/directori:
```
$cp /ruta/destí/
```
Moure archiu/directori:
```
$mv /ruta/destí/
```
Elimiar archiu/directori:
```
$rm /ruta/destí/
```
Llistar directori actual:
```
$ls o $dir
```
Mostrar configuració de les xarxes (IP's i dispositius):
```
$ifconfig
```
Mostrar taula de rutes:
```
$route -n
```
Mostrar els processos escoltant als ports TCP:
```
$netstat -tlnp
```
Mostrar els processos escoltant als ports UDP:
```
$netstat -ulnp
```

7.2.1 Expandir disc dur de la targeta microSD

Després de gravar l'imatge, es creen dues particions, una anomenada swap (intercanvi), i una altra principal on es troba tot el sistema operatiu. D'aquestes dues particions, una ocupa poca capacitat en MB(Mega Bytes) i la resta està sense definir.

Amb la següent comanda es configura la raspberry.

```
$raspi-config
```

La primera opció permet expandir la targeta de memòria a tota la seva capacitat. L'executarem per a poder instal·lar més programes a la targeta de memòria sense patir per l'espai en el disc dur.

La segona, canviar la contrasenya, és recomanable canviar-la per motius de seguretat.

La tercera opció són les opcions d'arrancada del sistema operatiu.

La quarta opció posa a la Raspberry Pi a l'espera de trobar una xarxa pera poder arrancar el sistema operatiu. No és necessari fer canvis.

La cinquena opció serveix per canviar la llengua i regió de la raspberry. Posar el que es desitgi.

La sisena opció habilita la càmera. No és el cas.

La setena opció és per registrar la raspberry. No és necessari.

La vuitena opció és el menú del rellotge. Res.

La novena opció són configuracions avançades, d'aquí necessitem habilitar els busos I2C i SCI.

L'ultima opció és informació sobre la raspberry.

És necessari fer un reiniciat al finalitzar.

7.2.2 Habilitar Internet per a poder instal·lar el programari necessari

Ara és necessària la connexió a Internet. Tenim dues opcions, connectar el cable Ethernet directament al router, o connectar l'adaptador wifi i configurar la xarxa per a poder establir la connexió.
La primer opció és la més senzilla i ràpida, un cop connectats a la nostra xarxa per cable, la connexió es configura automàticament pel DHCP del router. Si no es realitza la connexió, provar d'executar:

```
$ifdown eth0
```

I després:

```
$ifup eth0
```

És possible que sigui més complex de configurar, depenent del tipus a connexió que es disposi.

Per a configurar l'adaptador wifi per a establir connexió a Internet em de seguir els passos següents:

a) Connectar l'adaptador i executar:

```
$ifdown wlan0
```

b) Un cop connectat l'adaptador wifi, s'ha d'editar el fitxer interfaces a través de la línia de comandes, l'editor que ve de sèrie és el nano:

```
$nano etc/network/interfaces
```

S'ha de possar el SSID i la contrasenya de la nostra xarxa wifi com es mostra a continuació:

```
allow-hotplug wlan0
iface wlan0 inet dhcp
        wpa-ssid  NomXarxa
        wpa-psk   ''password''
```

Per sortir i guardar s'ha de clicar "ctrl esquerra+x" per sortir, "y" per acceptar, i "enter" per guardar.

c) Per acabar, s'han d'executar la següent comanda per a poder establir la connexió.

```
$ifup wlan0
```

d) Per a assegurar que tenim connexió a Internet podem fer un ping per a fer comprovacions:

```
$ping 8.8.8.8
```

7.2.3. Instal·lació i configuració del software necessari

Un cop tenim connexió a Internet, podem accedir al terminal de Raspbian, i actualitzar el SO fent servir la comanda:

```
$apt-get update
```

7.2.3.1. SSH

Instalarem l'SSH per a poder comunicar-nos a través d'altres màquines un cop finalitzat el muntatge.

```
$apt-get install ssh
```

No entrarem en profunditat a tractar la configuració del SSH, tan sols es creu convenient canviar el port per defecte del port 22 a un altre, per evitar intrusisme.

```
$/etc/ssh/sshd_config

# What ports, IPs and protocols we listen for
Port 2244
```

Reiniciar SSH i verificar des de una altra màquina:

```
$/etc/init.d/ssh restart

$ssh pi@raspberrypi -p 2244
```

7.2.3.2. Apache2

Apache és un servidor HTTP (de pàgines web) de codi obert multi-plataforma desenvolupat per Apache Software Foundation. Necessitem les característiques d'aquest programa i tan sols ens assegurem de que si ja hi és i s'actualitzi. HTTP sempre utilitza el port 80, per defecte el port que utilitza un navegador d'Internet. El programari del pack OMD utilitza el servidor Apache per crear el servidor web.

```
$apt-get install apache2 -y
```

Ja que hi som instal·larem un servei de correu electrònic alternatiu anomenat exim4 que és capaç en rebre emails enviats per la mateixa màquina.

```
$apt-get install exim4
```

7.2.3.3. Hostapd i isc-dhcp-server

Per a poder configurar l'adaptador USB com a Access Point, és necessari descarregar hostapd i isc-dhcp-server.

```
$apt-get install hostapd isc-dhcp-server
```

Per a configurar els arxius, en l'apartat 7.3.2.2 Access Point, s'explica en detall com fer-ho.

7.2.3.4. OpenVPN

Per instal·lar cal la comanda:

```
$apt-get install openvpn
```

Configurarem la VPN per a que el servidor tingui la IP: 10.8.0.1 i la Raspberry Pi utilitzi la IP: 10.8.0.6

Al directori de /etc/openvpn es troben les comandes per a generar la vpn:

```
$cd /etc/openvpn
```

Es construeix el certificat per a les claus de la VPN 'vpnpi'.

```
$./build-ca 'vpnpi'
```

Es creen les claus per al servidor VPN.

```
$./build-key-server 'server'
```

Es creen les claus per al clientVPN.

```
$./build-key 'client'
```

Es vinculen i xifren les claus VPN.

```
$./build-dh
```

Cal copiar tots els elements* de la carpeta /openvpn del server al servidor i deixar els arxius del client a la Raspberry Pi. En el nostre cas el servidor serà una màquina virtual situada a bohr.upc.es, domini de la UPC. Ens interessa que hi hagi un túnel directe a la Raspberry Pi des de un lloc públic per a poder accedir remotament via SSH (port 22) i consultar el port HTTP (port 80) per veure el Thruk. Llavors caldrà configurar el firewall o el filtratge NAT per a poder redirigir els ports.

*Els fitxer per al servidor són:
-server.conf
-'uptade-relov-conf'
-tot el directori 'easy-rsa'
-tot el directori 'keys'

'server.conf': (comentaris amb '#' i ';' al davant)

```
########################################################
# Sample OpenVPN 2.0 config file for                   #
# multi-client server.                                 #
#                                                      #
# This file is for the server side                     #
# of a many-clients <-> one-server                     #
# OpenVPN configuration.                               #
#                                                      #
# OpenVPN also supports                                #
# single-machine <-> single-machine                    #
# configurations (See the Examples page                #
# on the web site for more info).                      #
#                                                      #
# This config should work on Windows                   #
# or Linux/BSD systems.  Remember on                   #
# Windows to quote pathnames and use                   #
# double backslashes, e.g.:                            #
# "C:\\Program Files\\OpenVPN\\config\\foo.key"        #
#                                                      #
# Comments are preceded with '#' or ';'                #
########################################################
```

```
# Which local IP address should OpenVPN
# listen on? (optional)
#;local a.b.c.d
# Which TCP/UDP port should OpenVPN listen on?
# If you want to run multiple OpenVPN instances
# on the same machine, use a different port
# number for each one.  You will need to
# open up this port on your firewall.
port 1194

# TCP or UDP server?
;proto tcp
proto udp

# "dev tun" will create a routed IP tunnel,
# "dev tap" will create an ethernet tunnel.
# Use "dev tap0" if you are ethernet bridging
# and have precreated a tap0 virtual interface
# and bridged it with your ethernet interface.
# If you want to control access policies
# over the VPN, you must create firewall
# rules for the the TUN/TAP interface.
# On non-Windows systems, you can give
# an explicit unit number, such as tun0.
# On Windows, use "dev-node" for this.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
;dev tap
dev tun

# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel if you
# have more than one.  On XP SP2 or higher,
# you may need to selectively disable the
# Windows firewall for the TAP adapter.
# Non-Windows systems usually don't need this.
;dev-node MyTap

# SSL/TLS root certificate (ca), certificate
# (cert), and private key (key).  Each client
# and the server must have their own cert and
# key file.  The server and all clients will
# use the same ca file.
#
# See the "easy-rsa" directory for a series
# of scripts for generating RSA certificates
```

```
# and private keys.  Remember to use
# a unique Common Name for the server
# and each of the client certificates.
#
# Any X509 key management system can be used.
# OpenVPN can also use a PKCS #12 formatted key file
# (see "pkcs12" directive in man page).
ca keys/ca.crt
cert keys/server.crt
key keys/server.key  # This file should be kept secret

# Diffie hellman parameters.
# Generate your own with:
#   openssl dhparam -out dh1024.pem 1024
# Substitute 2048 for 1024 if you are using
# 2048 bit keys.
dh keys/dh2048.pem

# Configure server mode and supply a VPN subnet
# for OpenVPN to draw client addresses from.
# The server will take 10.8.0.1 for itself,
# the rest will be made available to clients.
# Each client will be able to reach the server
# on 10.8.0.1. Comment this line out if you are
# ethernet bridging. See the man page for more info.
server 10.8.0.0 255.255.255.0

# Maintain a record of client <-> virtual IP address
# associations in this file.  If OpenVPN goes down or
# is restarted, reconnecting clients can be assigned
# the same virtual IP address from the pool that was
# previously assigned.
ifconfig-pool-persist ipp.txt

# Configure server mode for ethernet bridging.
# You must first use your OS's bridging capability
# to bridge the TAP interface with the ethernet
# NIC interface.  Then you must manually set the
# IP/netmask on the bridge interface, here we
# assume 10.8.0.4/255.255.255.0.  Finally we
# must set aside an IP range in this subnet
# (start=10.8.0.50 end=10.8.0.100) to allocate
# to connecting clients.  Leave this line commented
# out unless you are ethernet bridging.
;server-bridge 10.8.0.4 255.255.255.0 10.8.0.50 10.8.0.100

# Configure server mode for ethernet bridging
```

```
# using a DHCP-proxy, where clients talk
# to the OpenVPN server-side DHCP server
# to receive their IP address allocation
# and DNS server addresses.  You must first use
# your OS's bridging capability to bridge the TAP
# interface with the ethernet NIC interface.
# Note: this mode only works on clients (such as
# Windows), where the client-side TAP adapter is
# bound to a DHCP client.
;server-bridge

# Push routes to the client to allow it
# to reach other private subnets behind
# the server.  Remember that these
# private subnets will also need
# to know to route the OpenVPN client
# address pool (10.8.0.0/255.255.255.0)
# back to the OpenVPN server.
;push "route 192.168.10.0 255.255.255.0"

# To assign specific IP addresses to specific
# clients or if a connecting client has a private
# subnet behind it that should also have VPN access,
# use the subdirectory "ccd" for client-specific
# configuration files (see man page for more info).

# EXAMPLE: Suppose the client
# having the certificate common name "Thelonious"
# also has a small subnet behind his connecting
# machine, such as 192.168.40.128/255.255.255.248.
# First, uncomment out these lines:
;client-config-dir ccd
;route 192.168.40.128 255.255.255.248
# Then create a file ccd/Thelonious with this line:
#   iroute 192.168.40.128 255.255.255.248
# This will allow Thelonious' private subnet to
# access the VPN.  This example will only work
# if you are routing, not bridging, i.e. you are
# using "dev tun" and "server" directives.

# EXAMPLE: Suppose you want to give
# Thelonious a fixed VPN IP address of 10.9.0.1.
# First uncomment out these lines:
;client-config-dir ccd
;route 10.9.0.0 255.255.255.252
# Then add this line to ccd/Thelonious:
#   ifconfig-push 10.9.0.1 10.9.0.2
```

```
# Suppose that you want to enable different
# firewall access policies for different groups
# of clients.  There are two methods:
# (1) Run multiple OpenVPN daemons, one for each
#     group, and firewall the TUN/TAP interface
#     for each group/daemon appropriately.
# (2) (Advanced) Create a script to dynamically
#     modify the firewall in response to access
#     from different clients.  See man
#     page for more info on learn-address script.
;learn-address ./script

# If enabled, this directive will configure
# all clients to redirect their default
# network gateway through the VPN, causing
# all IP traffic such as web browsing and
# and DNS lookups to go through the VPN
# (The OpenVPN server machine may need to NAT
# or bridge the TUN/TAP interface to the internet
# in order for this to work properly).
;push "redirect-gateway def1 bypass-dhcp"

# Certain Windows-specific network settings
# can be pushed to clients, such as DNS
# or WINS server addresses.  CAVEAT:
# http://openvpn.net/faq.html#dhcpcaveats
# The addresses below refer to the public
# DNS servers provided by opendns.com.
;push "dhcp-option DNS 208.67.222.222"
;push "dhcp-option DNS 208.67.220.220"

# Uncomment this directive to allow different
# clients to be able to "see" each other.
# By default, clients will only see the server.
# To force clients to only see the server, you
# will also need to appropriately firewall the
# server's TUN/TAP interface.
;client-to-client

# Uncomment this directive if multiple clients
# might connect with the same certificate/key
# files or common names.  This is recommended
# only for testing purposes.  For production use,
# each client should have its own certificate/key
# pair.
#
# IF YOU HAVE NOT GENERATED INDIVIDUAL
```

```
# CERTIFICATE/KEY PAIRS FOR EACH CLIENT,
# EACH HAVING ITS OWN UNIQUE "COMMON NAME",
# UNCOMMENT THIS LINE OUT.
;duplicate-cn

# The keepalive directive causes ping-like
# messages to be sent back and forth over
# the link so that each side knows when
# the other side has gone down.
# Ping every 10 seconds, assume that remote
# peer is down if no ping received during
# a 120 second time period.
keepalive 10 120

# For extra security beyond that provided
# by SSL/TLS, create an "HMAC firewall"
# to help block DoS attacks and UDP port flooding.
#
# Generate with:
#   openvpn --genkey --secret ta.key
#
# The server and each client must have
# a copy of this key.
# The second parameter should be '0'
# on the server and '1' on the clients.
;tls-auth ta.key 0 # This file is secret

# Select a cryptographic cipher.
# This config item must be copied to
# the client config file as well.
;cipher BF-CBC        # Blowfish (default)
;cipher AES-128-CBC   # AES
;cipher DES-EDE3-CBC  # Triple-DES

# Enable compression on the VPN link.
# If you enable it here, you must also
# enable it in the client config file.
comp-lzo

# The maximum number of concurrently connected
# clients we want to allow.
;max-clients 100

# It's a good idea to reduce the OpenVPN
# daemon's privileges after initialization.
#
# You can uncomment this out on
```

```
# non-Windows systems.
user nobody
group nogroup

# The persist options will try to avoid
# accessing certain resources on restart
# that may no longer be accessible because
# of the privilege downgrade.
persist-key
persist-tun

# Output a short status file showing
# current connections, truncated
# and rewritten every minute.
status openvpn-status.log

# By default, log messages will go to the syslog (or
# on Windows, if running as a service, they will go to
# the "\Program Files\OpenVPN\log" directory).
# Use log or log-append to override this default.
# "log" will truncate the log file on OpenVPN startup,
# while "log-append" will append to it.  Use one
# or the other (but not both).
log         openvpn.log
;log-append  openvpn.log

# Set the appropriate level of log
# file verbosity.
#
# 0 is silent, except for fatal errors
# 4 is reasonable for general usage
# 5 and 6 can help to debug connection problems
# 9 is extremely verbose
verb 9

# Silence repeating messages.  At most 20
# sequential messages of the same message
# category will be output to the log.
;mute 20
```

*Els fitxers per al client són:

-'client.conf'
-tot el directori 'keys'

'client.conf':

```
#############################################
# Sample client-side OpenVPN 2.0 config file #
# for connecting to multi-client server.     #
#                                            #
# This configuration can be used by multiple #
# clients, however each client should have   #
# its own cert and key files.                #
#                                            #
# On Windows, you might want to rename this  #
# file so it has a .ovpn extension           #
#############################################

# Specify that we are a client and that we
# will be pulling certain config file directives
# from the server.
client

# Use the same setting as you are using on
# the server.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
;dev tap
dev tun

# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel
# if you have more than one.  On XP SP2,
# you may need to disable the firewall
# for the TAP adapter.
;dev-node MyTap

# Are we connecting to a TCP or
# UDP server?  Use the same setting as
# on the server.
;proto tcp
proto udp

# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
remote ip_publica_router_wimax 1194
;remote my-server-2 1194

# Choose a random host from the remote
```

```
# list for load-balancing.  Otherwise
# try hosts in the order specified.
;remote-random

# Keep trying indefinitely to resolve the
# host name of the OpenVPN server.  Very useful
# on machines which are not permanently connected
# to the internet such as laptops.
resolv-retry infinite

# Most clients don't need to bind to
# a specific local port number.
nobind

# Downgrade privileges after initialization (non-Windows only)
;user nobody
;group nogroup

# Try to preserve some state across restarts.
#persist-key
#persist-tun

# If you are connecting through an
# HTTP proxy to reach the actual OpenVPN
# server, put the proxy server/IP and
# port number here.  See the man page
# if your proxy server requires
# authentication.
;http-proxy-retry # retry on connection failures
;http-proxy [proxy server] [proxy port #]

# Wireless networks often produce a lot
# of duplicate packets.  Set this flag
# to silence duplicate packet warnings.
mute-replay-warnings

# SSL/TLS parms.
# See the server config file for more
# description.  It's best to use
# a separate .crt/.key file pair
# for each client.  A single ca
# file can be used for all clients.
ca keys/ca.crt
cert keys/client1.crt
key keys/client1.key

# Verify server certificate by checking
```

```
# that the certicate has the nsCertType
# field set to "server".  This is an
# important precaution to protect against
# a potential attack discussed here:
#  http://openvpn.net/howto.html#mitm
#
# To use this feature, you will need to generate
# your server certificates with the nsCertType
# field set to "server".  The build-key-server
# script in the easy-rsa folder will do this.
ns-cert-type server

# If a tls-auth key is used on the server
# then every client must also have the key.
;tls-auth ta.key 1

# Select a cryptographic cipher.
# If the cipher option is used on the server
# then you must also specify it here.
;cipher x

# Enable compression on the VPN link.
# Don't enable this unless it is also
# enabled in the server config file.
comp-lzo

# Set log file verbosity.
verb 3

# Silence repeating messages
;mute 20
```

Iniciar openvpn.

```
$/etc/init.d/openvpn start
```

Un cop iniciada la VPN podem executar el servidor:

```
$~/servervpn.conf
```

I el client:

```
$~/clientvpn.conf
```

La commanda NAT a la Raspberry:

```
$iptables -t nat -A PREROUTING -o eth0 -p tcp -dport 8081 -j DNAT –-to
10.8.0.6:80
```

La commanda NAT al servidor bohr (primer instal·lar iptables):

```
$aptget install iptables
```

```
$iptables -t nat -A PREROUTING -o eth0 -p tcp -dport 8081 -j DNAT –-to 10.?.?.? (ip màquina virtual)
```

La VPN arrancarà sempre que tingui connexió a Internet i pugui connecta-se al servidor VPN.

7.2.3.5. Usb-modeswitch i sakis3g

Tan sols cal instal·lar-los, no cal configurar-los ja que s'invocaran conjuntament en un script.

```
$apt-get install usb-modeswitch sakis3g
```

7.2.3.6. Pack OMD i configuració del Nagios

Per a instal·lar el pack OMD en el sistema operatiu Raspbian s'han seguit els passos que mostra la web: labs.consol.de/repo/stable

Els passos són:

-Instal·lar GPG Key:

```
$gpg --keyserver keys.gnupg.net --recv-keys F8C1CA08A57B9ED7
gpg --armor --export F8C1CA08A57B9ED7 | apt-key add -
```

-Possar la web de reposicions al fitxer d'actualitzacions del sistema Raspbian Jessie:

```
$echo 'deb http://labs.consol.de/repo/stable/debian jessie main' >> /etc/apt/sources.list
```

-I actualitzar:

```
$apt-get update
```

Configuració del Nagios:

Per configurar el Nagios és necessari crear un usuari o "site" en el sistema operatiu Raspbian per poder gestionar el pack omd, l'anomenarem fapp:

```
$omd create fapp
```

Verifiquem:

```
$omd sites
SITE            VERSION
fapp            0.48 (default)
```

S'otorga un nom d'administrador i contrasenya per al monitoratge en navegador web.

Nom d'administrador: omdadmin
Contrasenya d'administrador: omd

Per entrar al 'site' fapp, sent *root*:

```
$su - fapp
```

Per defecte, no ve el motor del Nagios en marxa, si no el 'naemon'. Executar i habilitar el Nagios:

```
$omd config
```

Ara crearem els hosts i serveis necessaris, i les comandes que invoquen a les comprovacions.

En comptes de seguir l'esquema de fitxers de configuració, farem servir un arxiu únic, que utilitzi plantilles genèriques.

Els plugins ja existents es troben a /omd/sites/fapp/lib/nagios/plugins (USER1, per defecte).

Els plugins de Nagios normalment necessiten paràmetres d'entrada, però en el nostre cas no els hem utilitzat perquè tots els paràmetres són per defecte.

Els plugins o scripts creats, no necessiten paràmetres d'entrada, i es dipositen a /omd/sites/fapp/local/lib/nagios/plugins (USER 2).

Aquest és el fitxer únic de configuració, anomenat 'pi.conf', i ubicat a /omd/sites/fapp/etc/nagios/conf.d/:

```
define host {
 host_name            raspberrypi
 use                  generic-host
 address              127.0.0.1
}
define command {
 command_name         check_pi_temp
 command_line         $USER2$/check_pi_temp
}
define command {
 command_name         check_ae9_air_pressure
 command_line         $USER2$/simulations_pressure
}
define command {
 command_name         check_ae9_air_flow
 command_line         $USER2$/simulations_airflow
}
define command {
 command_name         check_ae9_air_capacity
 command_line         $USER2$/simulations_capacity
}
```

```
define command {
  command_name        check_ae9_power
  command_line        $USER2$/simulations_power
}
define service {
  use                 generic-service,srv-pnp
  host_name           raspberrypi
  service_description Temperature
  check_command       check_pi_temp
  contact_groups      administrators
  check_interval      4
  retry_interval      1
}
define service {
  use                 generic-service,srv-pnp
  host_name           raspberrypi
  service_description Air Pressure
  check_command       check_ae9_air_pressure
  contact_groups      administrators
  check_interval      4
  retry_interval      1
}
define service {
  use                 generic-service,srv-pnp
  host_name           raspberrypi
  service_description Air Flow
  check_command       check_ae9_air_flow
  contact_groups      administrators
  check_interval      2
  retry_interval      1
}
define service {
  use                 generic-service,srv-pnp
  host_name           raspberrypi
  service_description Air Capacity
  check_command       check_ae9_air_capacity
  contact_groups      administrators
  check_interval      2
  retry_interval      1
}
define service {
  use                 generic-service,srv-pnp
  host_name           raspberrypi
  service_description Power
  check_command       check_ae9_power
  contact_groups      administrators
  check_interval      4
```

```
 retry_interval          1
}
define service {
 use                     generic-service,srv-pnp
 host_name               raspberrypi
 service_description     Disk Usage
 check_command           check_local_disk!30%!10%!/
 contact_groups          administrators
 check_interval          2
 retry_interval          1
}
define service {
 use                     generic-service
 host_name               raspberrypi
 service_description     HTTP Service
 check_command           check_http
 contact_groups          administrators
 check_interval          2

 retry_interval          1
}
define service {
 use                     generic-service
 host_name               raspberrypi
 service_description     SSH Service
 check_command           check_ssh  #!p, si es canvia el port per defecte(22)
 contact_groups          administrators
 check_interval          2
 retry_interval          1
}
define service {
 use                     generic-service
 host_name               raspberrypi
 service_description     Ping Service
 check_command           check_ping
 contact_groups          administrators
 check_interval          2
 retry_interval          1
}
define contact{
 contact_name                    max
 use                             generic-contact
 service_notification_commands   notify-by-led, service-notify-by-email
 host_notification_commands      notify-by-led, host-notify-by-email
 email                           fapp
}
define contactgroup{
```

```
 contactgroup_name       administrators
 alias                   group of administrators
 members                 max
}

define command{
 command_name    notify-by-led
 command_line   /omd/sites/fapp/local/lib/nagios/plugins/turn_on_led
'$SERVICEDESC$'  '$SERVICESTATE$'
}
```

Aquest últim plugin (turn_on_led) és en realitat un script que s'encarrega d'encendre els leds quan els sensors estan en estat 'warning', posar-los en intermitent en estat 'critic' i apagar-los en cas d'una notificació en estat 'ok'. Necessita crear un contenidor per a dipositar les notificacions i que el programa en Python s'encarregui de la gestió dels leds, també genera un missatge de text per a poder consultar, igual que el correu electrònic local de la Raspberry.

Crearem el fitxer Python 'led-server.py' a: /omd/sites/local/bin
Crearem la pipe leds.cmd a:/omd/sites/fapp/tmp/run

Exemple per crear una 'pipe' o 'fifo':

```
$cd /omd/sites/fapp/tmp/run
$mkfifo pep
```

S'envia la notificació en format: "(on|off|blink) nº_de_led"

Per exemple:

```
$echo "blink 0" > pep
$cat pep
blink 0
```

Crearem un fitxer de configuració inicial del Nagios per assegurar que està creada la 'pipe' i que està executant-se el programa en Python led-server.py, a /omd/sites/fapp/etc/init_hooks.d, anomenat nagios-start-pre, que s'encarrega d'executar-se abans d'arrancar el Nagios.

'nagios-start-pre':

```
#!/bin/bash

LEDS_CMD_PIPE=/omd/sites/fapp/tmp/run/leds.cmd

if ! [ -p $LEDS_CMD_PIPE ]; then
mkfifo $LEDS_CMD_PIPE
fi

/omd/sites/fapp/local/bin/led-server.py &
```

Aquest és l'script 'turn_on_led':

```
#! /bin/bash
SERVICESTATE=$1
SERVICEDESC="$2"

#Es crea un missatge de text
echo "SERVICESTATE=$1 SERVICEDESC=$2" >>
/omd/sites/fapp/var/nagios/messages.txt

LEDS_CMD_PIPE="/omd/sites/fapp/tmp/run/leds.cmd"

if ! [ -p $LEDS_CMD_PIPE ]; then
        exit3
fi

#Determinar estat
if [ "$1" == "CRITICAL" ]; then
        state="blink"
elif    [ "$1" == "WARNING" ]; then
        state="on"
elif    [ "$1" == "OK" ]; then
        state="off"
else
        exit 0
fi

#Dipositar ordre a la pipe
case $2 in
        "Air Flow") echo "$state 0" > $LEDS_CMD_PIPE
         ;;
         "Air Pressure") echo "$state 1" > $LEDS_CMD_PIPE
        ;;
        "Air Capacity") echo "$state 2" > $LEDS_CMD_PIPE
        ;;
        "Power") echo "$state 3" > $LEDS_CMD_PIPE
        ;;
        "Temperature") echo "$state 4" > $LEDS_CMD_PIPE
        ;;
        "Disk Usage") echo "$state 5" > $LEDS_CMD_PIPE
        ;;
        "HTTP Service") echo "$state 6" > $LEDS_CMD_PIPE
        ;;
        "SSH Service") echo "$state 7" > $LEDS_CMD_PIPE
        ;;
esac
```

Finalment, l'script orientatiu que invoca al programa en C i llegeix la dada del sensor:

On $factor és el factor de conversió inventat per passar les dades de corrent en 'mA' a 'bars'.

```
#! /bin/bash

#Es crida al programa en c per obtenir el valor del sensor

diri2c="/home/pi/bcm2835/bcm2835-1.42/examples/i2c"

$factor

valor= tail ($diri2c/get_AI481Sv2 -p 1 -r 12 -g 1 -c 0x68)
# La sortida és "Code: 0xba (186d),  Current (mA): 4.0582"

pres=$valor*35

if  [ $pres -gt $crit ]; then
 RV=2
elif [ $pres -gt  $warn ]; then
   RV=1
else
   RV=0
fi

warn=625
crit=650
min=0
max=700

MSG[0]="OK"
MSG[1]="WARNING"
MSG[2]="CRITICAL"

echo -n "The pressure is $pres bars: ${MSG[$RV]} "
echo "| Pressure=$pres;$warn;$crit;$min;$max"

exit $RV
```

Per iniciar o reiniciar el nagios s'ha d'exectuar:

```
$omd start o restart
```

Accedir al Thruk:

Des d'un navegador d'Internet es pot accedir al Thruk, posant:

ip-de-la-raspberry/usuari/thruk (p.e. 127.0.0.1/fapp/thruk des de la mateixa Raspberry Pi)

Captura 4. Inici de sessió al Thruk

S'accedeix amb el nom d'usuari i la contrasenya atorgades en el moment de fer l'usuari omd.
Nom d'administrador: omdadmin
Contrasenya d'administrador: omd

Per crear les interfícies gràfiques(GUIs) on poder mostrar les dades, s'utilitzarà l'eina 'dashboard' del programa Thruk, que s'accedeix des de "Panorama view". Amb un xic de paciència, es pot crear un bon GUI a partir d'imatges estàtiques i tot tipus de gràfics predefinits en el programa. Es recomana anar pas a pas i tenir oberta la finestra de configuració dels elements fins que es finalitza el disseny de cada element(que es guarda clicant 'save'), ja que és possible que si es modifica un element, i no es guarda després, es modifica com a l'última actualització. Els GUIS creats es mostren a l'apartat 8.2. Resultats.

Des de l'interfície gràfica es pot accedir a totes les altres aplicacions de la fundació, cal habilitar la GUI 'welcome' a la configuració d'omd, executant:

```
$omd config
```

Per accedir al Thruk a través de la VPN en un navegador web: bohr.upc.es:8081/fapp/thruk

7.3. Configuració de les xarxes

7.3.1. Ethernet

Per configurar la xarxa de la Raspberry Pi a través de connexió Ethernet, s'ha de connectar el cable RJ45 al port Ethernet de la Raspberry, i si està connectat a un router amb DHCP ja es configurarà sol.

Si es vol configurar la xarxa per Ethernet es pot modificar el fitxer interfaces, el sistema operatiu nomena eth0 al dispositiu Ethernet.

```
$nano etc/network/interfaces
```

Segons les característiques de la xarxa , posar:

```
iface eth0 inet manual
        address 192.168.42.1
        netmask 255.255.255.0
```

I per finalitzar, executar:

```
$ifup eth0
```

7.3.2. Adaptador wifi

El sistema operatiu nomena wlan0 aquest dispositiu de xarxa.

7.3.2.1. Client dhcp

Aquest mode és el més utilitzat en la llar avui dia. El router wifi al qual ens connectem és qui s'encarrega d'atorgar IP i connexió a Internet automàticament. Tan sols es necessita saber la contrasenya i l'identificador SSID o nom de la xarxa, semblant a WLAN_XXX.

```
$nano etc/network/interfaces
```

Posar:

```
allow-hotolug wlan0
iface wlan0 inet dhcp
        wpa-ssid  SSID
        wpa-psk   ''password''
```

Executar:

```
$ifup wlan0
```

Per a accedir a la Raspberry Pi, cal connectar-se a la IP especificada al fitxer interfaces: 192.168.42.1.

7.3.2.2. Access Point

Aquest mode està creat per a poder-se connectar a la pròpia xarxa wifi que genera el dispositiu USB des de, per exemple, un mòbil tipus smartphone, una tablet, o qualsevol aparell portàtil.

Per a configurar l'adaptador wifi com a Access Point seguirem els següents passos:

Editar el fitxer dhpd.conf:

```
$nano /etc/dhcp/dhcpd.conf
```

Posar comentaris (#) al començament de les línies:

```
#option domain-name "example.org";
#option domain-name-servers ns1.example.org, ns2.example.org;
```

Treure el comentari a:

```
authoritative
```

Afegir al final del fitxer:

```
subnet 192.168.42.0 netmask 255.255.255.0{
        range 192.168.42.10 to192.168.42.50;
        option broadcast-address 192.168.42.255;
        option routers 192.168.42.1;
        default -lease-time 600;
        max-lease-time 7200;
        option domain-name "local";
        option domain-name-servers 8.8.8.8, 4.4.4.4;
}
```

Editar isc-dhcp-server:

```
$nano /etc/default/isc-dhcp-server
```

Posar: INTERFACES = ''wlan0'' a la línia que toca.

Editar fitxer de configuració de xarxes 'interfaces':

```
$nano etc/network/interfaces
```

Posar:

```
allow-hotolug wlan0
iface wlan0 inet static
        address 192.168.42.1
        netmask 255.255.255.0
```

Crear fitxer hostapd.conf, i posar:

```
$nano /etc/hostapd/hostapd.conf
```

```
# /etc/hostapd/hostapd.conf:
logger_syslog=-1
```

```
logger_syslog_level=1
logger_stdout=-1
logger_stdout_level=1
interface=wlan0
ctrl_interface=/var/run/hostap-wlan0
driver=nl80211
country_code=ES
ssid=Max_AP
wpa_passphrase=1234567890
hw_mode=g
channel=1
wpa=2
auth_algs=1
macaddr_acl=0
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
ignore_broadcast_ssid=0
ieee80211n=1
ieee80211d=1
preamble=0
disassoc_low_ack=1
ht_capab=[HT40+][SHORT-GI-40][GF][TX-STBC][RX-STBC1]
bssid=e8:4e:06:12:25:bd
# tunning
wmm_enabled=1
# Low priority / AC_BK = background
wmm_ac_bk_cwmin=4
wmm_ac_bk_cwmax=10
wmm_ac_bk_aifs=7
wmm_ac_bk_txop_limit=0
wmm_ac_bk_acm=0
# Normal priority / AC_BE = best effort
wmm_ac_be_aifs=3
wmm_ac_be_cwmin=4
wmm_ac_be_cwmax=10
wmm_ac_be_txop_limit=0
wmm_ac_be_acm=0
# High priority / AC_VI = video
wmm_ac_vi_aifs=2
wmm_ac_vi_cwmin=3
wmm_ac_vi_cwmax=4
wmm_ac_vi_txop_limit=94
wmm_ac_vi_acm=0
# Highest priority / AC_VO = voice
wmm_ac_vo_aifs=2
wmm_ac_vo_cwmin=2
```

```
wmm_ac_vo_cwmax=3
wmm_ac_vo_txop_limit=47
wmm_ac_vo_acm=0

beacon_int=100
dtim_period=2
max_num_sta=255
rts_threshold=2347
fragm_threshold=2346
eap_server=0
eapol_key_index_workaround=0
```

On driver és l'identificador del USB i ssid i wpa_passphrase són el nom de la xarxa i la contrasenya per accedir a la xarxa des de qualsevol dispositiu.

Especificar configuració del hostapd, trobar la lìnia DAEMON_CONF.

```
$nano /etc/default/

DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

Arrancar isc-dhcp-server i hostapd

```
$/etc/init.d/hostapd start
$/etc/init.d/isc-dhcp-server start
```

És possible que existeixi algun tipus d'error en executar aquestes commandes, millor utilitzar:

```
$service hostapd start
$service isc-dhcp-server start
```

Es pot accedir a la xarxa que genera el dispositiu wifi idènticament que una connexió wifi estàndard. Cal connectar-se a la SSID ja mencionada, i posar la contrasenya. Per a accedir a la Raspberry Pi, cal connectar-se a la IP especificada al fitxer interfaces: 192.168.42.1.

Si és cal utilitzar el mode conjuntament amb Ethernet per a tenir connexió Internet a la subxarxa creada per l'Access Point, cal fer reenvio (IP forwarding), executant:

```
$echo 1 > /proc/sys/net/ipv4/ip_forward
```

7.3.3. Mòdem 3G

Per a que la Raspberrypi pugui reconèixer un dispositiu usb com a mòdem, necessitem que s'executi el programa usb-modeswitch, llavors, el sistema operatiu nomena wwan0 aquest dispositiu.
El programa/script sakis3g servirà per establir connexió a Internet per 3G.
Crearem un Script que executi els programes en ordre, però primer necessitem saber l'identificador del nostre mòdem USB, consultem la llista que serà quelcom semblant i ens quedem amb el codi:

```
$ lsusb
```

```
Bus 001 Device 011: ID 12d1:1464 Huawei Technologies Co., Ltd.
Bus 001 Device 006: ID 148f:5372 Ralink Technology, Corp. RT5372 Wireless
Adapter
Bus 001 Device 005: ID 0566:3004 Monterey International Corp. Genius KB-29E
Bus 001 Device 004: ID 046d:c050 Logitech, Inc. RX 250 Optical Mouse
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.
SMSC9512/9514 Fast Ethernet Adapter
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

I creem l'script:

```
if [ "$1" == start ]; then
        # Check if usb is connected
        lsusb | grep -E "12d1:(1464|1521)" > /dev/null
         if [ $? -ne 0 ]; then
        echo "Error: Modem not present"
        exit 1
          fi

  usb_modeswitch -W -I -v 0x12d1 -p 0x1521 -M
55534243123456780000000000000011060000000000000000000000000000

   sakis3g connect --console  --debug MODEM="12d1:1464"
USBINTERFACE=0 SIM_PIN=2444 APN="airtelnet.es" APN_USER="vodafone"
APN_PASS="vodafone" DIAL="*99#"

elif [ "$1" == stop ]; then
        sakis3g disconnect --console  MODEM="12d1:1464"
USBINTERFACE=0 SIM_PIN=2444 APN="airtelnet.es" APN_USER="vodafone"
APN_PASS="vodafone" DIAL="*99#"
else
        echo "$0 (start | stop)"
fi
```

Es crea connexió 3G automàticament. Per poder contemplar la possibilitat de connecta-se remotament via SSH, cal connectar-se a partir de la IP pública 169.254.12.135 que atorga Vodafone.

7.4. Configuració Piface Digital 2

Nota: connectar la targeta perifèrica Piface amb la Raspberrypi apagada.

Si no s'han habilitat els busus SCI prèviament, torna a raspi-config:

```
$raspi-config
```

Per veure la llista d'opcions avançades de configuració, seleccioneu l'opció 8 Opcions avançades.
Trieu l'opció A5 SCI. Posar "Sí"
Seleccioneu "OK" i després "Finalitzar"

Si es desitja, es pot instal·lar l'emulador de PiFace Digital, executeu l'ordre:

```
$apt-get install-python3 piface digital emulator
```

Escrigui S quan se li demani.

Exemple:

```
$python3 /usr/share/doc/python3-pifacedigitalio/examples/blink.py
```

Iniciar un nou intèrpret de Python teclejant python o python3 en un terminal. Escriu les ordres de Python després que aparegui el símbol >>>. Les comandes s'executarà un a un a mesura que els introdueix, llevat que estan contingudes dins de sentències de control, com ara 'while' i 'if'.

Aquesta és una gran manera de provar fragments de codi i explorar les diferents característiques dels productes PiFace.

Per crear un tipus de programa en python3 ells proposen:

```
$touch NomdelPrograma.py
```

Però també es pot obrir aquest arxiu amb el teu editor de text favorit, per exemple:

```
$nano NomdelPrograma.py
```

S'explica la programació en Python més endavant, tan sols s'ha creat un programa per a gestionar els leds segons les notificacions rebudes pel Nagios.

7.5. Connexió via SSH

Com s'ha explicat, via SSH podem accedir remotament al dispositiu. Per això hem de tenir el servei SSH actiu. Si no s'ha activat el servei des del menú de configuració inicial, serà necessari accedir a un terminal per executar les ordres;

```
$/etc/init.d/ssh start
$/etc/init.d/ssh stop
$/etc/init.d/ssh restart
```

Per establir connexió, si em canviat el port, per defecte és el 22. x:x:x:x és la IP de la raspberry. La màquina amb la que es realitza la connexió SSH ha de pertànyer a la mateixa xarxa o poder accedir a través d'una ip pública.

```
$ssh pi@x:x:x:x -p 'nºport'
```

Per a copiar arxius remotament és necessari la utilització de SecureCopyProtocol, dintre ssh:

Sintaxis SecureCopyProtocol:

```
scp [-12346BCEpqrv] [-c cipher] [-F ssh_config] [-i identity_file] [-l limit] [-o ssh_option] [-P port] [-S program] [[user@]host1:]file1 ... [[user@]host2:]file2
```

Copiar un arxiu local a un destí remot:

```
$scp /ruta/al/arxiu-origen usuari@ordinador:/ruta/al/directori-desti/
```

Copiar tots els arxius d'un tipus específic (.ext).

```
$scp /ruta/al/directori/*.ext usuari@servidor:/ruta/al/directori/
```

Copiar tots els arxius d'un directori recursivament al servidor remot:

```
$scp -r /home/user/html/* jane@host.example.com:/home/jane/backup/
```

7.6. Programació

A part de la programació shellscripting que ja s'ha mencionat, aquí es mostrarà el codi utilitzat per programar en Python la PifaceDigital, i en C, la placa I²C.

7.6.1. Programació en Python

El programa que s'ha creat en Python serveix per a estar executat en segon pla (background) a l'escolta d'un contenidor on es dipositen les notificacions amb el format "(on|off|blink) nº_de_led". El programa s'encarrega d'encendre, apagar o fer intermitència els leds de la Piface Digital2.

Crearem l'script amb sheebang de Python 'led-server.py' a /omd/sites/local/bin:

```
#!/usr/bin/python
# vim: set ts=4 sw=4 tw=0 noet :
import re, select, os, time

# Codi per executar amb la RPi i la piface:
import pifacedigitalio
pdf = pifacedigitalio.PiFaceDigital()

# Emular enum type
def enum(*sequential, **named):
enums = dict(zip(sequential, range(len(sequential))), **named)
return type('Enum', (), enums)
```

```
# Definir led state com 'on' 'off' or 'blinking'
led_state = enum('on','off','blink')


# Inicialitza LEDS STATUS a 'off state'
LEDS_STATUS = [ led_state.off, led_state.off, led_state.off,
                led_state.off, led_state.off, led_state.off,
                led_state.off, led_state.off]

# Període de Blinking (intermitència)
blink_time = 0.0

# Canvia l'estat (on, off, blink) del led corresponent
def set_led_state(led_num, new_state):
        global LEDS_STATUS, pdf
        LEDS_STATUS[int(led_num)] =  new_state
        if new_state == led_state.blink:
                pdf.leds[int(led_num)].toggle()
                pass
        elif new_state == led_state.on:

                pdf.leds[int(led_num)].turn_on()
                pass
        elif new_state == led_state.off:
                pdf.leds[int(led_num)].turn_off()
                pass

# Aquesta funció s'invoca cada BLINK_PERIOD segons
# i comprova quins leds estàn blinking
def blink_leds():
        global LEDS_STATUS, blink_time, pdf
        # This var only to test blinking period
        now = time.time()
        for i in range(len(LEDS_STATUS)):
                if LEDS_STATUS[i] == led_state.blink:
                        pdf.leds[i].toggle()
                        print "%f toogle led %d" % (now-blink_time, i)
        blink_time = now

pdf = pifacedigitalio.PiFaceDigital()

# Definir 'blink period' en segons
BLINK_PERIOD=1
outputs = [ ]
```

```
to = time.time() + BLINK_PERIOD
timeout = BLINK_PERIOD

if __name__ == '__main__':
	while True:
		fd = os.open("pep",os.O_RDONLY | os.O_NONBLOCK)
		rp = os.fdopen(fd)
		inputs = [ rp ]
		rlist, wlist, xlist = select.select(inputs , outputs , inputs,
		timeout)
		if not (rlist or wlist or xlist) :


		blink_leds()
		now = time.time()
		if now >= to:
			timeout = BLINK_PERIOD
			to = now + timeout
		else :
			timeout = to - now
		continue
	#Llegir de la pipe cada cop que està disponible una notificació
	result = rp.read()
	cmd = re.search("(on|off|blink) ([0-7])", result)
	if cmd is not None:
		action =  cmd.group(1)
		if action == 'on':
			action = led_state.on
		elif action == 'off':
			action = led_state.off
		elif action == 'blink':
			action = led_state.blink
		led = cmd.group(2)
		set_led_state(led, action)
		print_leds_status()
		now = time.time()
		if now >= to:
			timeout = BLINK_PERIOD
			to = now + timeout
		else :
			timeout = to - now
```

7.6.2. Programació en C

El programa en C necessari per establir connexió amb la placa I²C requereix de coneixements per a modificar-lo. El programa s'encarrega de comunicar-se amb la placa i fer una lectura del valor de corrent que li otorga la placa. Cada cop que s'invoca crea una escriptura en el bus que la placa contesta.

```
#include <bcm2835.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdint.h>
#include <unistd.h>

#define MAX_LEN 32

#define AI418S_ADDRESS              0x68
#define AI418S_A0                   0x00    /*A0*/
#define AI418S_A1                   0x00    /*A1*/
#define AI418S_A2                   0x00    /*A2*/

#define AI418_CONFIGURATION_REGISTER   0xD0    /*Write Address*/

/* Configuration Registe Bits */
#define AI418S_RDY                  0x80    /*Ready bit*/

/* Configuration Registe Bits */
#define AI418S_RDY                 0x80    /*Ready bit*/
#define AI418S_C1                  0x40    /*Channel Selection bit 1 */
#define AI418S_C0                  0x20    /*Channel Selection bit 0 */
#define AI418S_OS                  0x10    /*One Shot measurement mode*/
#define AI418S_S1                   0x08   /*Sample Rate Selection bit 0*/
#define AI418S_S0                   0x04   /*Sample Rate Selection bit 1*/
#define AI418S_G1                   0x02   /*PGA Gain Selection bit 1*/
#define AI418S_G0                   0x01   /*PGA Gain Selection bit 0*/

#define K_TIMES                     100

uint16_t clk_div = BCM2835_I2C_CLOCK_DIVIDER_148;

uint8_t slave_address = 0x00;
uint8_t channel;
uint8_t precision;
uint8_t gain;
uint8_t vc;
void usage(char *);
int comparse(int,  char **);
```

```
//***********************************************************************
//  comparse: Parse the command line and return EXIT_SUCCESS or
//EXIT_FAILURE
//    argc: number of command-line arguments
//    argv: array of command-line argument strings
//***********************************************************************


int comparse(int argc, char **argv) {
   int opt;
   opterr = 0;
   channel=0;
   gain=0;
   precision=0;
   vc=255;
   while ((opt = getopt (argc, argv, "p:r:g:vc")) != -1)
      switch (opt) {
         case 'p':
            channel = atoi(optarg);
            if ((channel < 1) || (channel > 4)) {
               return EXIT_FAILURE;
            }
            channel--;
            break;
          case 'r':
               precision = atoi(optarg);
               switch (precision) {
                        case 12:
                        case 14:
                        case 16:
                        case 18:
                        break;;
                        default:
                                return EXIT_FAILURE;
                }
                precision=precision/2-6;
                break;
          case 'g':
                gain = atoi(optarg);
                switch (gain) {
                        case 1: gain = 0; break;;
                        case 2: gain = 1; break;;
                        case 4: gain = 2; break;;
                        case 8: gain = 3; break;;
                        default:
                                return EXIT_FAILURE;
                 }
```

```
        break;

      case 'v':
            if (vc != 255) {
                  return EXIT_FAILURE;
            }
           vc=1;
           break;
      case 'c':
            if (vc != 255) {
                  return EXIT_FAILURE;
           }
            vc=0;
           break;
      case '?':
            if (optopt == 'p')
      fprintf (stderr, "Option -p requires a port number 1-4.\n");
           else if (optopt == 'r')
      fprintf (stderr, "Option -r requires the DAC resolution (12,14,16,18
      bits).\n");
            else if (optopt == 'g')
      fprintf (stderr, "Option -g requires the DAC gain (1,2,4,8).\n");
           else if (isprint (optopt))
      fprintf (stderr, "Unknown option `-%c'.\n", optopt);
           else
      fprintf (stderr,"Unknown option character `\\x%x'.\n",optopt);
           return EXIT_FAILURE;
      default:
           return EXIT_FAILURE;
  }

  int pending_args = argc - optind;

  if (pending_args != 1) {
     return EXIT_FAILURE;
  }
   sscanf(argv[optind], "%x", &slave_address);
   return EXIT_SUCCESS;
}

void usage(char *progname) {
printf("%s -c channel -p precision -g gain Slave_Address\n", progname);
printf("options:\n");
printf("  -p    port from 1 to 4\n");
printf("  -r    DAC resolution (12,14,16,18) bits\n");
printf("  -g    DAC gain 1,2,4,8\n");
printf("  (-v|-c)    For voltage or current measure\n");
```

```
    printf("\n   Default values: -p 1 -r 12 -g 1 -c\n");
    }

int main(int argc, char **argv) {
     uint8_t data;

      // parse the command line
      if (comparse(argc, argv) == EXIT_FAILURE)  usage (argv[0]);
      if (!bcm2835_init()) return 1;

      // I2C begin if specified
      bcm2835_i2c_begin();
      // Handle channel and precision to build configuration register
       channel<<= 5;
       precision<<=2;

       uint8_t bytes_to_read = precision == 12 ? 4 : 3;
       uint8_t rxBuffer[4] = {0};
       uint8_t txBuffer[1] = {0};

      // Build configuration register value
        txBuffer[0] = channel | precision | gain | AI418S_RDY ;

        bcm2835_i2c_setSlaveAddress(AI418S_ADDRESS);
        bcm2835_i2c_setClockDivider(clk_div);

        data = bcm2835_i2c_write(txBuffer, 1);

      switch (data) {
            case BCM2835_I2C_REASON_ERROR_NACK:
                  printf("Received a NACK while reading AI418 data\n");
                  return EXIT_FAILURE;
                  break;
            case BCM2835_I2C_REASON_ERROR_CLKT:
                  printf("Received a Clock Stretch Timeout while reading AI418
                  data\n");
                  return EXIT_FAILURE;
                  break;
            case BCM2835_I2C_REASON_ERROR_DATA:
                  printf("Not all data is sent / received while reading AI418
                  data\n");
                  return EXIT_FAILURE;
                  break;
            }

             if ( !(rxBuffer[bytes_to_read - 1] & AI418S_RDY) ) {
             //fprintf(stdout,"Tried %d times read of measure\n",k+1);
```

```
            break;
        }
        k++;
      }
        if ( k == K_TIMES) {
       fprintf(stdout,"Reached K_TIMES and not RDY\n");
          return 1;
       uint32_t result = 0;
        for (uint8_t i=0;i<bytes_to_read-1;i++) {
           result<<=8;
           result = result | rxBuffer[i];
         }
         if ((vc == 0) || (vc == 255)) {
         printf("Code: 0x%x (%id),  Current (mA): %2.4f\n", result, result,
      (float)4*result/(2048*(1<<(precision/2)))*(2.048/(1<<gain))*180/33);
         }
         if (vc == 1) {
         printf("Code: 0x%x (%id),  Voltage (V.): %2.4f\n", result, result,
      (float)result/(2048*(1<<(precision/2)))*(2.048/(1<<gain))*180/33);

        }
         bcm2835_i2c_end();
        bcm2835_close();
        //printf("... done!\n");
        return 0;
      }
```

Per compilar el codi, utilitzar la comanda:

```
$gcc – std=gnu99 get_AI481Sv2.c -o get_AI481Svs - lcbm2835
```

Per realitzar proves es prova amb la comanda:

```
$/home/pi/bcm2835/bcm2835-1.42/examples/i2c/get_AI481Sv2 -p 1 -r 12
-g 1 -c 0x68
```

-p de port (canal)
-r de rate (velocitat, per defecte, 12)
-g de gain (amplificació, per defecte, 1)
-c de corrent (v de voltatge)

La sortida, per exemple, és "Code: 0xba (186d), Current (mA): 4.0582"

7.7. Conclusions

Finalment, El nostre sistema està correctament configurat i preparat per a poder establir connexió a Internet o a una xarxa privada i pot ser configurat per manteniment o actualització a través de la connexió remota. Pot comprovar els sensors de qualsevol tipus de màquina i generar notificacions en cas de situació d'emergència o urgència. Les proves s'han realitzat a cada pas, i com a resultat, es mostren captures de pantalla de les interfícies gràfiques creades per a la visualització de les dades dels sensors. Aquestes dades s'actualitzen en períodes de temps curts i el programari utilitzat és completament flexible a l'hora de configurar-se, i permet obtenir els resultats desitjats.

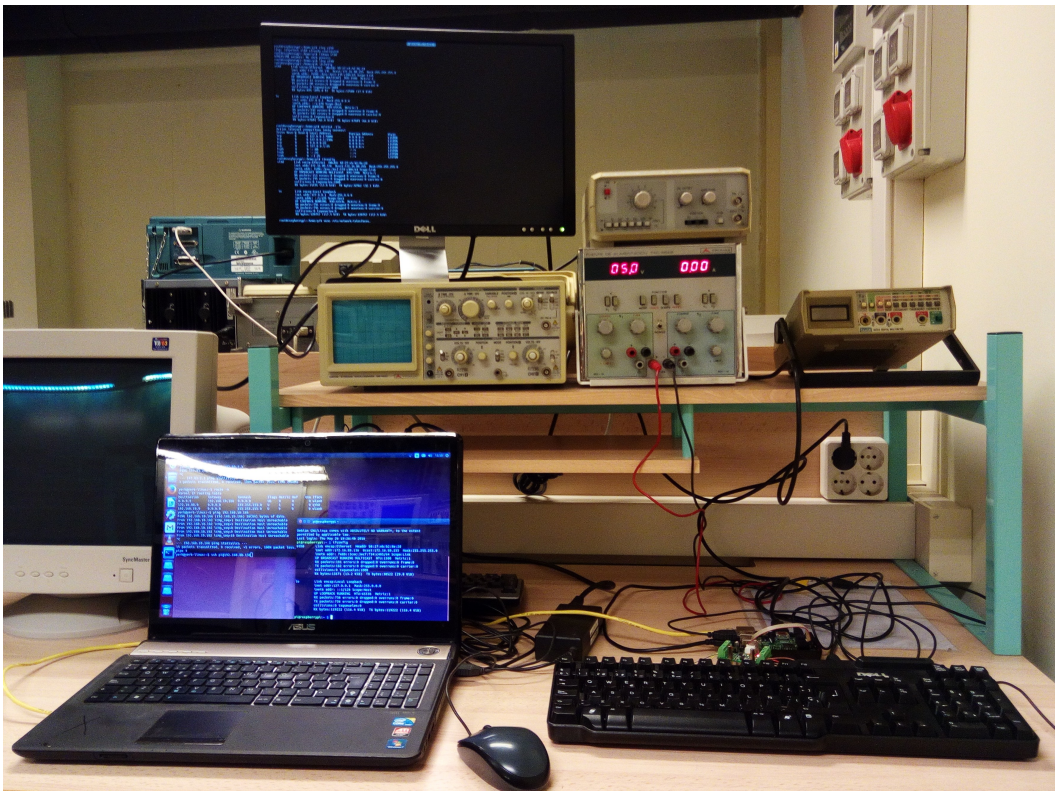
8. PROVES Y RESULTATS

8.1. Proves

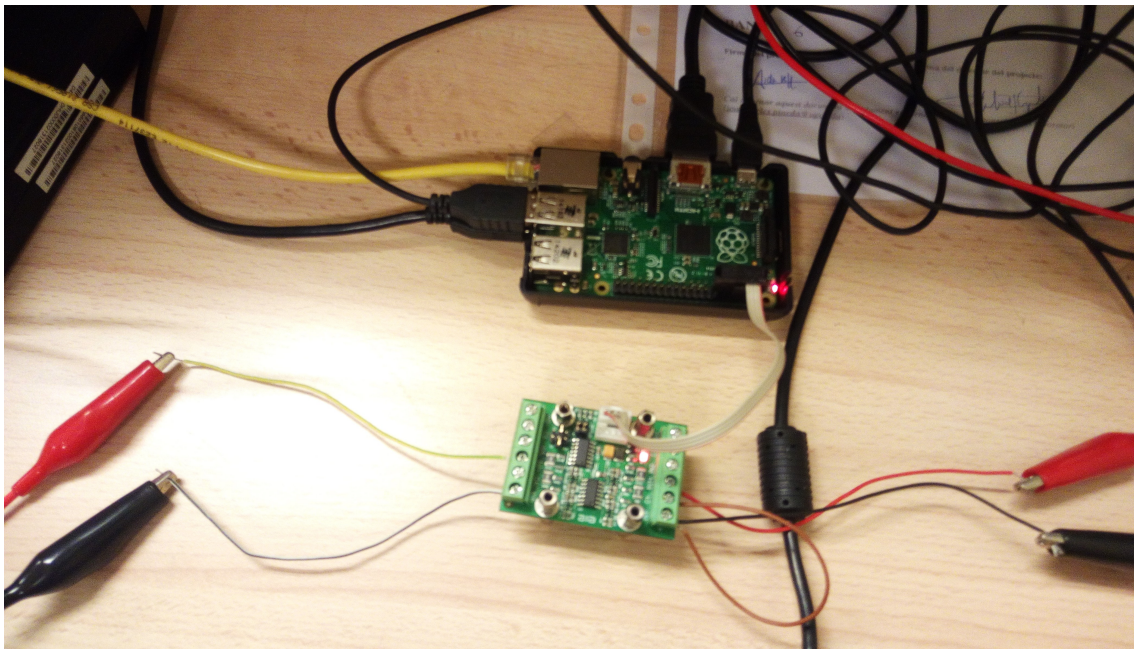
Per a la realització de les proves, s'ha procedit a observar el comportament dels sensors simulats en una interfície gràfica (GUI) creada amb l'eina 'dashboard' del programa Thruk, que s'accedeix des de "Panorama view" en el navegador d'Internet. S'han creat diferents GUIs ja que s'intenta mostrar com de flexible és el programa a l'hora de crear les interfícies gràfiques.

El muntatge detallat al desenvolupament del projecte s'ha realitzat en el període de quatre mesos, comprovant cada error abans d'avançar. En els dos primers mesos es va realitzar la configuració de la Raspberry Pi. El tercer mes es van programar el hardware adicional per separat, ja que per unir les dues plaques (La PiFace Digital i la I2C-AI418S) es necessiten unes soldadures en els pins que no s'utilitzen de la PiFace per a utilitzar el busos que requereix la placa I2C. En l'últim mes s'ha realitzat la memòria i comprovant tots els erros finals.

Les simulacions al Laboratori de Projectes van resultar satisfactòries i es va constatar que el programa té una resolució de microAmpers.



Imatge 12. Muntatge de la simulació amb fonts d'alimentació.



Imatge 13. Muntatge de la simulació amb fonts d'alimentació.

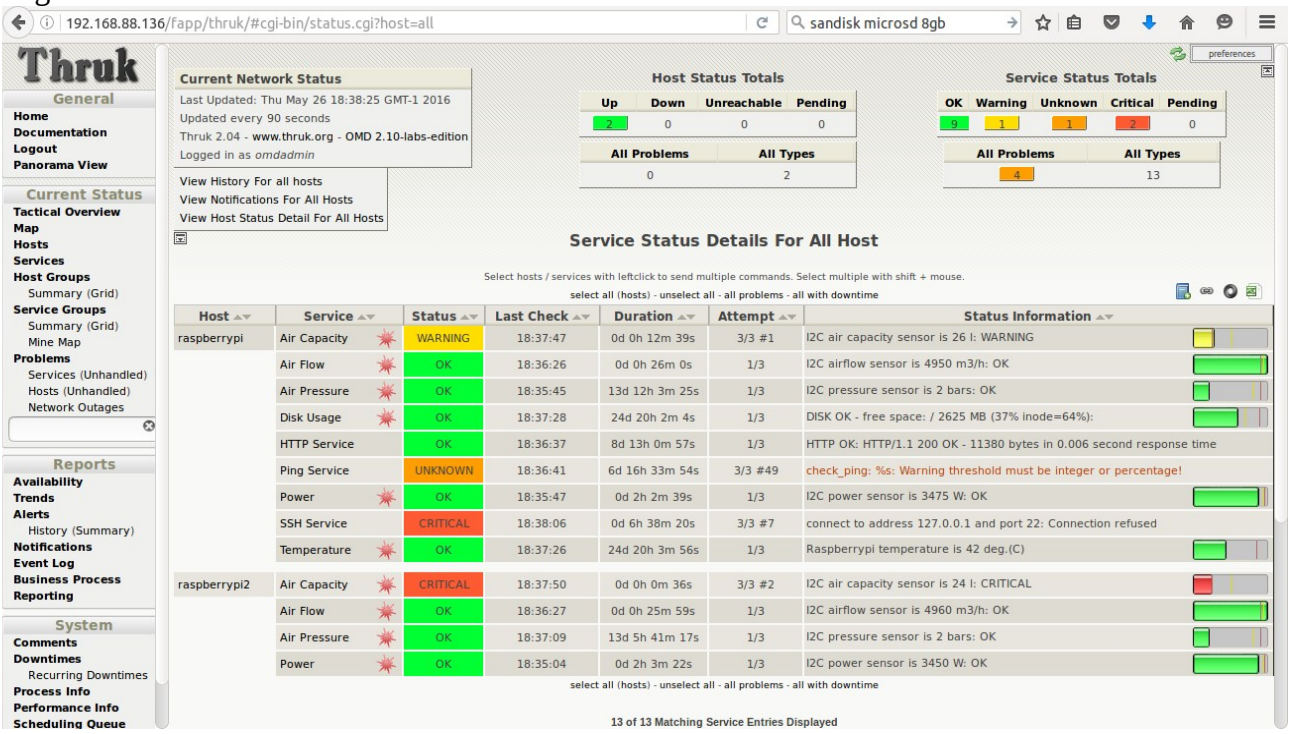
8.2. Resultats

El sistema de notificacions ha creat més de 1800 correus electrònics locals en dos mesos, i estan guardats a la Raspberry Pi, en el contenidor de correu electrònic de l'usuari fapp. Per exemple un dels últims és:

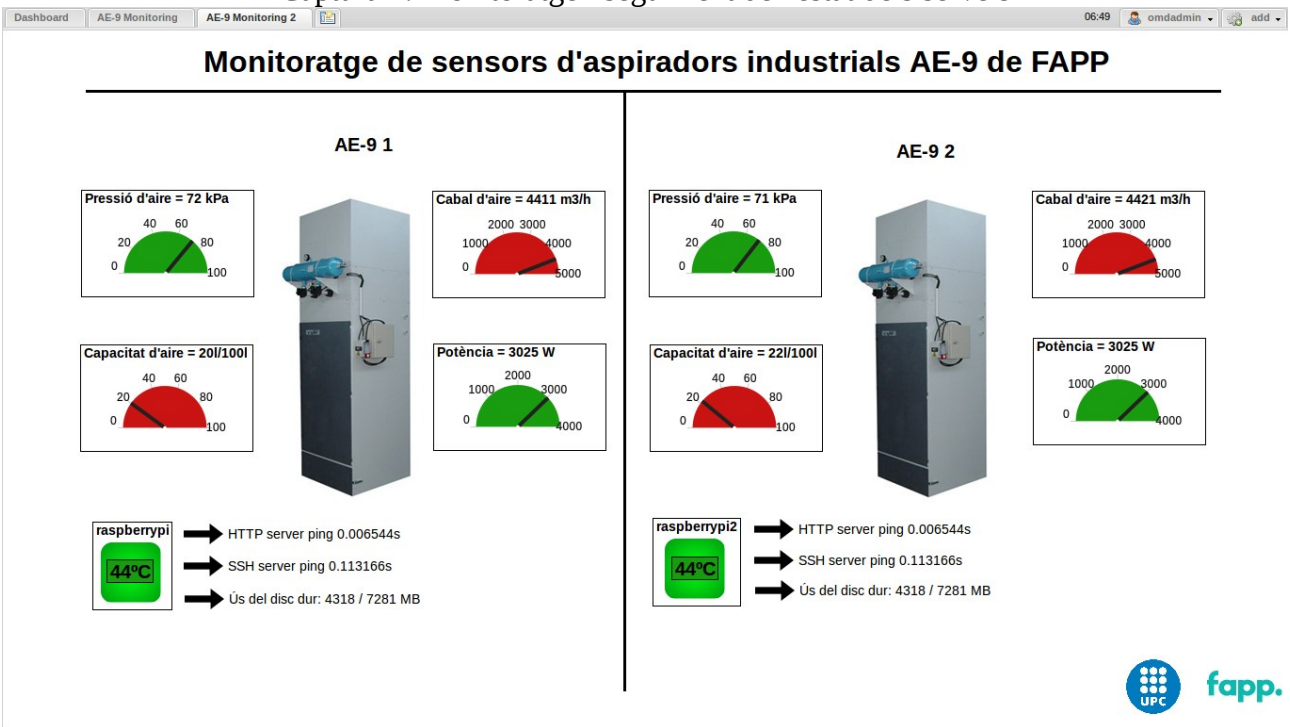
```
        Message 1772:
        From fapp@raspberrypi Sat May 28 16:48:51 2016
        Return-path: <fapp@raspberrypi>
        Envelope-to: fapp@raspberrypi
        Delivery-date: Sat, 28 May 2016 16:48:51 +0100
        Date: Sat, 28 May 2016 16:48:51 +0100
        To: fapp@raspberrypi
        Subject: *** PROBLEM *** raspberrypi / Air Capacity is WARNING
        User-Agent: Heirloom mailx 12.5 6/20/10
        Content-Type: text/plain; charset=us-ascii
        From: OMD site fapp <fapp@raspberrypi>
        Status: R

        --SERVICE-ALERT----------------
        -
        - Hostaddress:     127.0.0.1
        - Hostname:        raspberrypi
        - Service:         Air Capacity
        - - - - - - - - - - - - - - - - -
        - State:           WARNING
        - Date:            2016-05-28 16:48:48
- Output:           Air capacity sensor is 28 l: WARNING
```

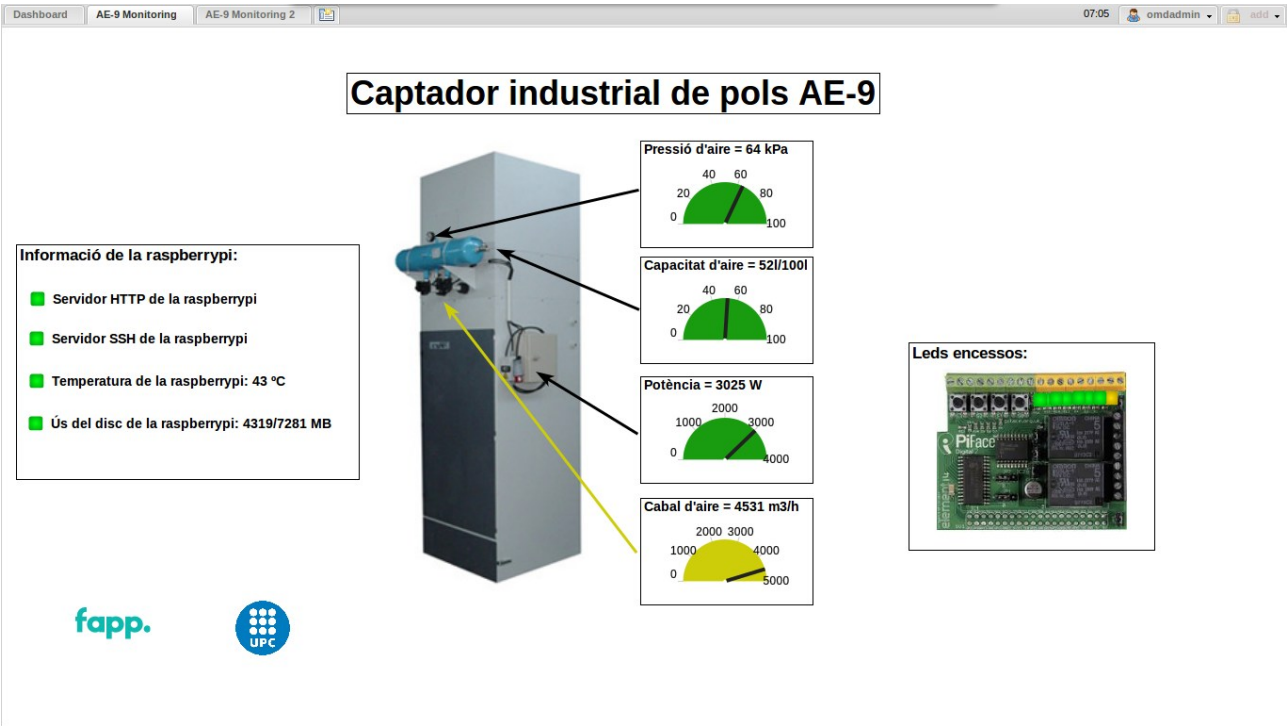
Seguidament es mostren les GUIS del Thruk.



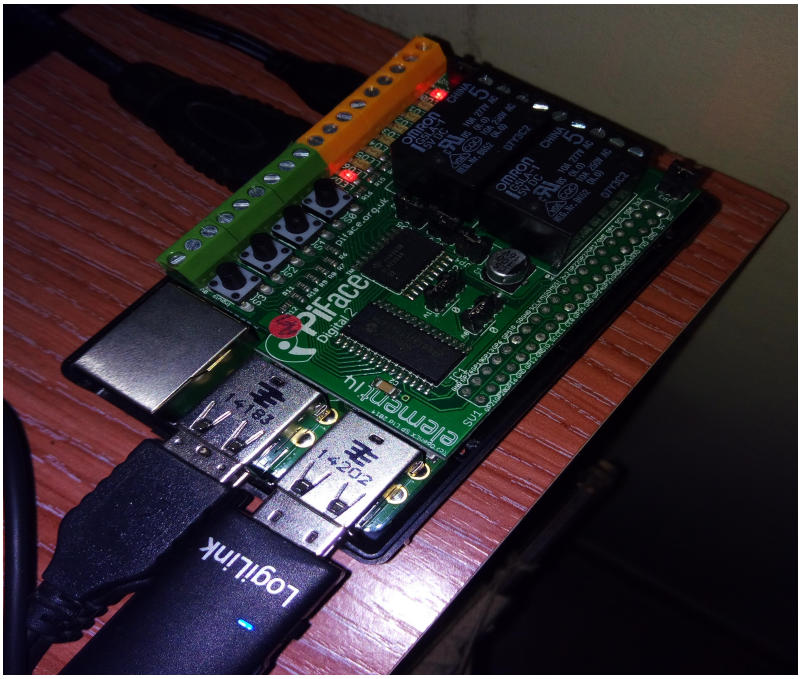
Captura 4. Monitoratge i seguiment de l'estat dels serveis



Captura 5. Monitoratge i seguiment de sensors de dos captadors de pols diferents.



Captura 6. Monitoratge i seguiment de sensors i l'estat dels serveis mostrat per led a la Piface.



Imatge 14. Leds encesos amb les notificacions.

9. CONCLUSIONS

9.1.Conclusions

Com ja s'ha mencionat, aquest projecte intenta mostrar un exemple de com es pot utilitzar un ordinador de placa reduïda per gestionar i monitoritzar les dades d'uns sensors situats dintre qualsevol màquina. Les captures de pantalla realitzades amb el programari utilitzat, faciliten la comprensió final del projecte. A més de crear mètodes de notificació alternatius al correu electrònic o al SMS, s'han creat les notificacions visuals amb leds, això dóna idea de com s'amplia el ventall de possibilitats i de com es pot explotar més aquest programari. Una altre possibilitat seria habilitar la sortida de so de la Raspberry Pi per a notificar amb alarmes sonores, un cas que no s'ha dut a terme ja que el captador de pols AE-9 radia molta pressió sonora (72dBA) i el sistema d'alarma hauria d'estar situat lluny de la màquina, o exercir molt més soroll, casos que s'haurien d'estudiar *in situ.*

El sistema data-logger dissenyat en aquest projecte millora i facilita el seguiment i monitoratge d'equips situats remotament, i resulta molt útil per a dur a terme un manteniment de totes les maquines que ho requereixin. Per exemple, un fabricant pot fer servir aquest sistema de baix cost per a fer realitzar el seguiment de les màquines distribuïdes als seus clients, i proporcionar telemanteniment (manteniment a distància).

El projecte permet moltes millores, ampliacions, optimitzacions i noves funcionalitats, cosa que permet adaptar-lo per funcionar conjuntament amb altres aparells i per augmentar així el nombre de possibilitats d’ús.

10. BIBLIOGRAFÍA

En ordre de consulta:

Captador de pols AE-9:

http://www.fappsa.com/

Informació de RaspberryPi :

https://www.raspberrypi.org/
http://www.codeproject.com/
https://learn.adafruit.com/series/learn-raspberry-pi
http://www.internetdelascosas.cl/2013/10/13/configurando-motion-con-la-camara-de-raspberry-pi/
http://www.tuelectronica.es/tutoriales/raspberry-pi/capturar-fotos-hd-con-el-modulo-camara-raspberry-pi.html
https://learn.adafruit.com/adafruit-raspberry-pi-lesson-1-preparing-and-sd-card-for-your-raspberry-pi/what-next
http://raspberryparatorpes.net/tag/webcam-raspberry-pi/

Informació de la targeta Piface:

http://www.piface.org.uk/guides/Install_PiFace_Software/

Informació de la placa I2C-AI418S 4-20ma DAC:

https://www.raspberrypi.org/forums/viewtopic.php?f=44&t=75582
https://pinout.xyz/pinout/i2c
https://www.ereshop.com/shop/free/I2C-AI418S_SHEET.pdf
https://www.ereshop.com/shop/free/I2C-AI418S_SCH.pdf
https://www.ereshop.com/shop/free/MCP3424.pdf

Informació de Raspbian :

http://www.raspbian.org/
http://raspberrypihq.com/how-to-turn-a-raspberry-pi-into-a-wifi-router/
http://blog.desdelinux.net/permisos-basicos-en-gnulinux-con-chmod/
https://learn.adafruit.com/adafruits-raspberry-pi-lesson-3-network-setup
http://raspberryparatorpes.net/empezando/raspi-config-2015-configuar-raspbian-paso-a-paso/

Informació del programa SSH:

http://www.openssh.com/txt/ssh-rfc-v1.3.txt
http://crysol.org/es/ssh-public-key
http://www.hypexr.org/linux_scp_help.php
http://www.tecmint.com/scp-commands-examples/

Informació del Pack OMD:

labs.consol.de/repo/stable
http://omdistro.org
http://www.nagios.org
http://www.nagiosexchange.org

Document en pdf:
W. Barth.
Nagios, System and Network Monitoring.
Open Source Press GmbH, first edition, 2006.
ISBN: 1-59327-070-4.

Informació de VPN:

https://openvpn.net/
https://openvpn.net/index.php/access-server/docs/quick-start-guide.html#asclient

Informació de la programació:

http://es.wikibooks.org/wiki/El_Manual_de_BASH_Scripting_B%C3%A1sico_para_Principiantes/Comandos_b%C3%A1sicos_de_una_shell
http://thales.cica.es/rd/glinex/practicas-glinex05/manuales/bash/practica.pdf
http://www.freeos.com/guides/lsst/
https://www.python.org/download/releases/3.4.1/
http://www.tutorialspoint.com/cprogramming/

Informació per al mode Access Point (HostAPD & isc-dhcp-server):

http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=444
http://w1.fi/hostapd/
https://wiki.debian.org/DHCP_Server

Webs consultades per a crear el pressupost:

Raspberry Pi Model B Motherboard:
http://www.amazon.com/Raspberry-Pi-756-8308-Motherboard-RASPBRRYPCBA512/

PI FACE Digital 2:
https://www.element14.com/community/docs/DOC-69001/l/piface-digital-2-for-raspberry-pi

I2C-AI418S 4-20ma DAC:
https://www.ereshop.com/shop/index.php?main_page=product_info&cPath=143&products_id=805&zenid=e1907d1df91d315557d6f705b53dcf8c

Accessoris extres per a la connexió de la placa I2C:

https://www.ereshop.com/shop/index.php?main_page=product_info&products_id=798&zenid=972789df869e3fd96c6fab33415d28a5
https://www.ereshop.com/shop/index.php?main_page=product_info&products_id=812&zenid=972789df869e3fd96c6fab33415d28a5

Raspberry Pi and PiFace Case:

http://www.amazon.com/Raspberry-PiFace-Case-Enclosure-BOX/dp/B00IJZV47G/ref=pd_bxgy_147_2?ie=UTF8&refRID=03XZN84QBM4JC15TK3WG

Logilink N150 WiFi adapter antenna:

http://www.conrad-electronic.co.uk/ce/en/product/990399/Logilink-N150-WiFi-adapter-antenna

SanDisk Ultra 8GB Class 10 UHS-I MicroSDHC Memory Card:

http://www.amazon.com/SanDisk-microSDHC-Standard-Packaging-SDSQUNC-032G-GN6MA/dp/B010Q57T02/ref=sr_1_2?s=pc&ie=UTF8&qid=1462467703&sr=1-2&keywords=microsd

Mòdem 3G USB model k4505 de Huawei comercialitzat per Vodafone:

http://consumer.huawei.com/en/?pinfoId=2210&directoryId=2462&treeId=462
http://bandaancha.eu/articulos/duelo-velocidad-internet-movil-hspa-6975

*Opcional: descàrrega de firmware per al mòdem 3G USB k4505:

bandaancha.eu/store/devices/huawei/k4505/VMC-Lite_9.4.4.17702.rar

Informació dels sensors:

http://www.digikey.com/es/resources/conversion-calculators/conversion-calculator-pressure
http://www.sensores-de-medida.es/sensing_sl/SENSORES-Y-TRANSDUCTORES_35/Sensores-de-presi%C3%B3n_107/Sensores-de-presi%C3%B3n-industriales_108/Sensor-de-presi%C3%B3n-salida-puente-wheatstone-AEP-TP1_141.html