

Estació meteorològica en xarxa.

Estudiant

Artur Nebot Garrigó

“Enginyeria Tècnica d'Informàtica de Gestió”

Consultor

Jordi Bécares Ferrés

11 de juny de 2013

Índex

Estació meteorològica en xarxa.....	1
Introducció.....	5
Justificació.....	5
Descripció.....	6
Objectius.....	7
Enfocament.....	8
Fase de recerca i investigació.....	8
Fase de documentació.....	8
Fase d'acoblament i muntatge.....	8
Fase de codificació.....	9
Fase de testeig.....	9
Planificació.....	10
Planificació inicial.....	10
Factors que han modificat la planificació real.....	11
Principals tasques afectades en la planificació.....	13
Recursos Emprats.....	14
Recursos de maquinari emprats.....	14
Recursos de programari emprats.....	17
Productes obtinguts.....	18
Sistema de captura de dades.....	18
Client de recepció de les dades.....	19
Memòria del projecte.....	20
Descripció dels propers capítols.....	20
Antecedents.....	21
Estat de l'art.....	21
Estudi de mercat.....	22
Sistemes encastats.....	22
Comunicació entre els dispositius.....	27
Descripció funcional.....	29
Sistema total.....	29
Mota.....	31
Inicialització de la xarxa.....	33
Actualitzar configuració.....	34
Enviament.....	35
Comprovació de les alarmes.....	36
Centre de recepció de dades.....	37
Configuració de la mota.....	38
Mostra de les dades capturades.....	39
Alimentació de les dades.....	39
Descripció detallada.....	40
Estructura de dades i missatges.....	40
Mida del buffer de dades.....	43
Sensors.....	44

Altres apartats	50
Joc de proves	50
Testeig dels sensors	50
Proves de connexió.....	50
Ampliacions possibles	52
Viabilitat tècnica.....	53
Protocol d'intercanvi de dades obert.....	53
Transparència entre la implementació entre el capturador i el receptor.....	55
Capturar les magnituds físiques.....	56
Grau d'autonomia.....	56
Tolerància a fallades.....	57
Valoració econòmica	58
Conclusions	60
Bibliògrafa	61
Annexos.....	62

Agraïments.

*A tots els que han fet possible la realització d'aquest projecte,
en especial atenció a la família amics i a la parella
per tot el seu suport i sobretot la seva paciència.*

Introducció

Justificació

Des del principi dels temps s'ha volgut conèixer i controlar els fenòmens meteorològics del nostre entorn, per aquesta raó l'obtenció i anàlisi de dades resulta una eina clau per a comprendre, descobrir i fins i tot preveure l'evolució dels factors meteorològics del nostre entorn.

La constant evolució tecnològica i la reducció de cost dels dispositius electrònics ens permeten trencar amb moltes de les limitacions amb les que ens trobàvem temps enrere, i poder posar a l'abast de qualsevol dispositiu avançats de monitorització de dades a un cost fàcilment assequible.

Actualment és habitual trobar-nos amb dispositius d'àmbit domèstic capaços de capturar dades atmosfèriques molt més avançades que els clàssics termòmetres, tals com el nivell de lluminositat, humitat, pressió atmosfèrica etc..

La present memòria descriu el procés de desenvolupament d'un sistema informàtic autònom amb capacitat per a poder capturar algunes dades del nostre entorn i poder-les comunicar mitjançant un protocol d'intercanvi de dades obert a un sistema receptor per tal de realitzar un posterior anàlisi.

Aquest sistema pot ser instal·lat en gairebé qualsevol lloc que compleixi l'únic requisit de cobertura d'una xarxa sense fils.

Descripció

El projecte desenvolupat permet la creació d'un sistema de monitorització de diferents magnituds atmosfèriques de baix consum adaptable a diferents necessitats.

El nucli del sistema, esta basat en una petita placa encastada on hi ha instal·lats diversos sensors, per tal de capturar magnituds físiques de l'atmosfera. Aquestes magnituds, convertides a dades son reportades periòdicament mitjançant una connexió sense fils, i son consumides per un sistema receptor que s'encarregarà de processar-les.

Per tal de maximitzar la integració del nostre sistema amb qualsevol sistema d'informació de emmagatzematge i processat de dades, s'ha apostat per utilitzar un protocol estàndard d'intercanvi de dades anomenat JSON i una interfície oberta a través del protocol HTTP, això amplia el seu àmbit d'us eliminant restriccions i facilitant la seva integració en altres sistemes independentment de la seva naturalesa.

Alhora també s'ha desenvolupat a mode d'exemple un petit client web que permet mostrar les dades i configurar les opcions de la mota.

Objectius

El nostre sistema requereix el compliment d'una sèrie de requisits per tal de poder ser considerat un projecte viable i útil.

- El protocol d'intercanvi de dades ha de ser obert
- El sistema de recepció ha de ser completament transparent a la mota, canvis en la seva implementació no poden afectar al procés de captura i reportatge de dades.
- Ha de proporcionar una interfície de comunicació oberta per tal de poder assolir amb èxit el punt anterior
- Ha de ser capaç de reportar una sèrie de magnituds físiques
 - Humitat ambiental
 - Temperatura ambiental
 - Lluminositat
- Ha de proporcionar un alt grau d'autonomia

Poder ser situada en qualsevol punt sense problemes d'instal·lació d'infraestructura específica, és a dir calbejat d'alimentació i de comunicacions.

- Tolerància a caigudes de xarxa
- Cal que proporcioni una sèrie de mecanismes per tal de minimitzar la pèrdua de dades.

Pot donar-se el cas en determinades condicions es perdin dades, però aquestes condicions s'han de trobar fitades i controlades.

- Tolerància a caigudes de xarxa

Degut a la seva ubicació el sistema ha de tolerar petites fallades de xarxa, degut a una mala cobertura o condicions adverses.

Enfocament

La realització del projecte ha passat per diferents cicles i fases, la majoria d'elles s'han realitzat seqüencialment però d'altres han sigut transversals al desenvolupament del projecte.

Fase de recerca i investigació

Fase inicial de coneixement de la tecnologia necessària per a la implementació del projecte, investigació de les característiques tècniques de la mota, sensors necessaris etc..

Aquesta fase ha permès identificar quines dades es requerirà i investigar en les metodologies i tecnologies necessàries per a obtenir-les.

Fase de documentació

La fase de documentació ha estat transversal a tot el procés de desenvolupament, iniciant-se en la fase inicial de recerca i investigació, continuant durant tot el procés de codificació i finalitzant en la darrera fase de testeig

Ha permès deixar constància clara de tot el procés realitzar i permetre a futurs desenvolupament tindre una pauta i un punt de partida.

Fase d'acoblament i muntatge

Fase de muntatge dels dispositius necessaris per l'assoliment del projecte, instal·lació dels sensors i els mòduls de connectivitat externa.

Ens ha permès construir el dispositiu físic que després es programarà per a complir amb els requeriments del projecte.

Fase de codificació

Fase d'escriptura del programari necessari per a dur a terme tota la lògica funcional del nostre sistema. Junt amb la fase d'acoblament ens permet construir el conjunt de dispositius finals i proporcionar-li la funcionalitat desitjada.

Fase de testeig.

Fase de comprovació que el projecte final assoleix els requisits marcats inicialment amb el rendiment desitjat.

Resulta una fase crítica per tal de comprovar que el sistema realment funciona i compleix amb els requeriments esperats.

Planificació

Planificació inicial

La planificació inicial feta a l'arrencada del projecte dividia el desenvolupament del projecte en diverses fases, on cada una agrupava un conjunt de tasques ben definides.

Una primera fase de definició de les estructures de dades, per a processar i emmagatzemar les dades que més endavant es requerirà per a la implementació de la lògica del projecte.

Finalitzada la primera fase, es comença a desenvolupar el projecte per la part més crítica, el desenvolupament de la mota. Començant per el muntatge del dispositiu i l'escriptura dels controladors per a cada sensor. Un cop finalitzada la tasca d'escriptura dels controladors ja es pot començar a desenvolupar la lògica per a l'enviament de les dades.

Finalitzat el desenvolupament de la mota caldrà escriure el client que permetrà consumir les dades i visualitzar-les per tal que l'usuari final pugui interactuar amb el sistema.

La tasca de documentació es desenvolupa en paral·lel a l'escriptura de codi, enviant al tutor un primer esquema de la memòria i desenvolupant-la segons les anotacions, comentaris i correccions fetes.

Finalitzat el desenvolupament de client i servidor tenim una primera versió del sistema que caldrà provar per tal de detectar falles en el sistema, aquest procés es portarà a terme durant una setmana.

El procés de testeig detecta errors en el sistema, ens reservem doncs una setmana per tal de corregir els errors detectats.

Nombre	Fecha de inicio	Fecha de fin
• Fase inicial	14/04/13	15/04/13
• Sistema encastat	16/04/13	17/05/13
• Escriptura dels controladors	16/04/13	6/05/13
• Controlador dispositiu humitat	16/04/13	22/04/13
• Controlador dispositiu temperatura	23/04/13	29/04/13
• Controlador dispositiu lluminositat	30/04/13	6/05/13
• Serialització de dades	7/05/13	10/05/13
• Enviament de les dades	11/05/13	17/05/13
• Servidor	18/05/13	23/05/13
• Rebuda de les dades	18/05/13	21/05/13
• Deserialització de les dades	22/05/13	22/05/13
• Visualització de dades	23/05/13	23/05/13
• Fase de testeig	18/05/13	24/05/13
• Correcció	25/05/13	31/05/13
• Documentació	25/05/13	8/06/13

Figura 1 - plantejament inicial

Factors que han modificat la planificació real

- **Equivocació en la compra dels sensors**

La compra dels sensors va presentar problemes, els sensors que vaig comprar inicialment eren massa petits i resultaven impossibles de soldar amb un soldador d'estany domèstic.

- **A la planificació inicial mancaven algunes fases**

- Baix consum

Dins l'apartat de desenvolupament de la mota, mancava una tasca específica centralitzada en la implementació del baix consum. La lectura de la documentació i la manca d'una orientació clara sobre com implantar-lo van allargar la tasca de desenvolupament de la mota.

- Muntatge de la mota.

Dins l'apartat de la mota mancava una tasca específica de muntatge de la mota, soldatge dels sensors etc..

- **Pocs coneixements d'electrònica**

Els pocs coneixements previs d'electrònica i la manca d'experiència en sistemes encastats va fer que algunes tasques s'allarguessin més del inicialment plantejat.

- **Dificultats en el muntatge**

La poca experiència prèvia en soldadura va requerir més temps del pensant en el muntatge de la mota.

- **Ampliació dels requisits**

Implementació del sistema d'alarmes, va requerir 4 dies més dels inicialment plantejats.

- **Problemes amb el '*workspace*'**

El *workspace* base utilitzat per a desenvolupar el projecte tenia problemes en l'enllaçat del projecte, cosa que m'ha obligat a refer-lo en un parell d'ocasions, després de realitzar moltes proves.

- **Compaginació del treball final de carrera amb la vida laboral**

El desenvolupament del projecte en paral·lel i fora del a jornada laboral a minvat el temps de dedicació diari que s'ha pogut dedicar al projecte,

Principals tasques afectades en la planificació

Al llarg del desenvolupament de tot el projecte hi ha hagut dos grans moments de bloqueig on per falta d'experiència i d'informació m'he trobat completament aturat.

Desenvolupament dels controladors dels sensors

He resultat una de les tasques més dures en el llarg del desenvolupament del projecte, la falta d'exemples dins la pròpia documentació del fabricant de la placa, els pocs coneixements d'electrònica i la falta d'hàbit en el desenvolupament a baix nivell van fer que aquest punt resultes especialment difícil.

Finalment, després de moltes setmanes de llegir documentació i proves vaig aconseguir tenir una versió funcional dels lectors, però va demorar la planificació un parell de setmanes enrere.

Baix consum

No havia programat mai res en l'àmbit de la optimització d'energia i el meu plantejament inicial de la implementació era incorrecta.

A més a més, inicialment la implementació del baix consum no funcionava per un problema amb el projecte que contenia el codi del baix consum que feia que la tasca que s'encarregava de posar la mota en '[sleep mode](#)' mai s'executés, va caldre refer tot el projecte i perdre uns quants dies de proves per a detectar i solucionar el error.

Recursos Emprats

Per al desenvolupament del projecte s'ha requerit d'una sèrie de recursos necessaris tan de maquinari com de programari per tal de fer possible la realització del projecte.

Els recursos de maquinari ens han proporcionat els dispositius necessaris per tal d'implementar les funcions requerides, i els de programari eines i utilitats per a la implementació de la lògica de l'aplicació.

Recursos de maquinari emprats

- Mota LPC1769

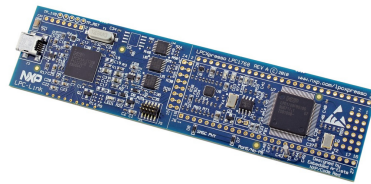


Figura 2 - Placa LPC1768

Mota principal per al desenvolupament del projecte del sistema captador de dades, les seves funcions principal són la lectura de dades dels sensors, l'enviament de les dades i la detecció de les alarmes.

- Sensor de temperatura i humitat SHT11 (Acoblament Parallax)

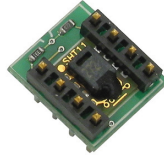


Figura 3- Sensor SHT11 integrada en el mòdul parallax

Sensor digital per a la captura de la temperatura i humitat relativa, forma part del sistema captador de dades. El seu funcionament en detall s'explica en capítols posteriors.

- Cèl·lula fotovoltaica FW-150



Figura 4 - Cèl·lula fotovoltaica FW-150

Proporciona la informació sobre la lluminositat de l'entorn, va connectada al sistema captador de dades. El seu funcionament en detall s'explica en capítols posteriors.

- Interfície de xarxa WiFly



Figura 5 - Mòdul de xarxa WiFly

Mòdul de xarxa per a la mota, proporciona la capacitat de comunicació sense fils a la mota, bàsic per l'intercanvi de dades entre el centre receptor de dades i el centre captador.

- Mòdul USB CP2102 principalment per a tasques de debug.



Figura 6 - Interfície CP2102

Proporciona un COM virtual y una interfície cap a la UART TTL, útil per a mostrar dades i per a *debug* de la mota.

- Taula de *protopitat*



Figura 7 - Taula de proves sense soldar 'Protoboard'

Requerit per a connectar fàcilment els sensors i altres dispositius a la mota sense necessitat de soldadura, útil per a proves de connectivitat i testeig abans de definir l'estructura definitiva.

- Ordinador d'escriptori per al desenvolupament

Ha estat la principal eina que s'ha utilitzat per a l'escriptura de la lògica del sistema, la programació de les motes del client i la redacció de la documentació.

Adicionalment també s'ha utilitzat una placa 'Raspberry pi' i un *driver* ja implementat per al sensor SHT11 per a fer el testeig del *driver* del sensor desenvolupat.

Recursos de programari emprats

Per al desenvolupament del projecte ha calgut la utilització de certes eines de programari.

En el següent llistat descriurem els recursos de programari utilitzats per a dur a terme el projecte.

- IDE de desenvolupament LPCXpresso (basat en Eclipse)
- Sistema operatiu FreeRTOS
- Llibreria JSMN
Ha permès serialitzar les dades de les estructures internes utilitzades a el protocol d'intercanvi de dades JSON.
- Llibreries CMIS
- Llenguatge de programació PHP per a la plana web
- Servidor web Apache 2.2.20
- MongoDB db version v2.0.8
Base de dades utilitzada per el client de demo.
- Microsoft Word per a l'escriptura de la memòria
- Microsoft Visio per al a creació dels esquemes.
- Git per al control de versions i històric de canvis.

Productes obtinguts

El sistema complet esta format per tres productes, un sistema captador de dades amb capacitat per a detectar les magnituds físiques mitjançant els sensors, un sistema de recepció per a consumir, emmagatzemar i processar les dades, i finalment la documentació del projecte.

Sistema de captura de dades

S'ha creat un dispositiu capaç de capturar una sèrie de magnituds físiques cada cert període de temps, emmagatzemar-lo temporalment i anar reportant-lo a un centre receptor de dades segons uns intervals configurables.

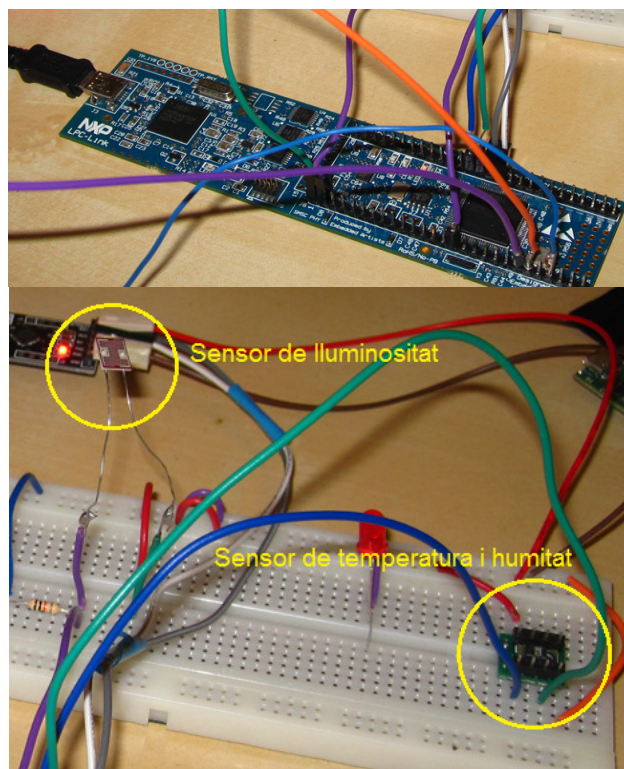


Figura 8 -Fotografies del prototipus i part del seu assemblatges

Artur Nebot Garrigó

Les magnituds atmosfèriques actualment suportades per el sistema captador són :

- Temperatura ambiental
- Humitat
- Lluminositat

El període entre cada captura de dades i entre cada enviament són configurables mitjançant el 'Client de recepció de les dades'.

El sistema implementa un mode de baix consum, on quan no hi ha dades a capturar o dades per a enviar desconnecta el rellotge i la connectivitat exterior permeten l'estalvi d'energia.

Client de recepció de les dades

Per tal de proporcionar persistència a les dades generades i permetre la interacció amb l'usuari s'ha desenvolupat un sistema receptor de dades .Aquest centralitza les tasques de recepció i control, proporcionant la configuració inicial al sistema de captura de dades i notificant a la mota els canvis en la configuració.

El client de recepció de dades consisteix servei web que proporciona una interfície d'alimentació de dades i obtenció de la configuració i una plana web que permet a l'usuari interactuar amb el sistema, visualitzar les dades informades i configurar els paràmetres del mòdul captador de dades.

Principals característiques del client:

- Proporciona una interfície web per tal que l'usuari pugui interactuar amb el sistema
- Facilita a l'usuari la tasca de visualització de les dades
- Permet a l'usuari la configuració dels intervals de temps entre lectures i entre enviaments
- Permet la configuració de les alarmes.
- Proporciona la capacitat de persistència de les dades, emmagatzemant de forma permanent les dades en una base de dades pròpia per al posterior tractament.

Artur Nebot Garrigó

Aquest programari client requereix d'una certa infraestructura per a poder funcionar

- Servidor web amb suport per a php y afegit per a la connectivitat amb la base de dades mongodb
- Instància de dades mongodb en execució

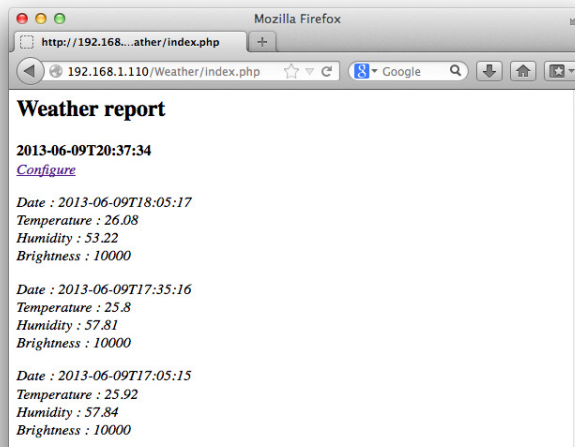


Figura 6 - Plana web de visualització de dades

Memòria del projecte.

Alhora també s'ha creat el present document, on s'explica detalladament tot el procés de desenvolupament del sistema, aquest inclou ; els requisits del sistema desenvolupat, les metodologies aplicades per al desenvolupament, la seva planificació, els detalls tècnics de la implementació etc..

Descripció dels propers capítols

Feta la introducció, definits els objectius del projecte i llistats els recursos utilitzats, en els propers capítols s'explicarà tota la lògica funcional del projecte entrant en detall en la implementació dels sectors crítics del projecte.

Antecedents

Estat de l'art

Un sistema encastat és un sistema de computació dissenyat per a satisfer unes funcions específiques on el consum, la mida del dispositiu i la connectivitat son factors clau dins les característiques del producte.

La principal diferència entre aquests i els ordinadors generals es que els ordinadors satisfan unes necessitats molt genèriques mentre els sistemes encastats son fabricats i programats per a acomplir únicament unes funcions determinades.

Els sistemes encastats s'inicien amb les investigacions de la companyia IBM cap als anys 80, actualment han evolucionat moltíssim i en molts casos tenen característiques que els fan molt propers als orientadors de propòsit general, trencant les limitacions clàssiques dels sistemes encastats.

Els sistemes encastats acostumen a ser circuits electrònics amb tots els components (memòria, processador...) o la seva majoria integrats dins la placa. Tenen unes mides reduïdes i el seu consum es troba molt fitat, essent força reduït.

Un factor força important dins els sistemes encastats és la comunicació, per això acostumen a incorporar diferents bussos per a poder-hi connectar una ampla varietat de dispositius tals com sensors, disc , dispositius de xarxa etc..

Ens podem trobar sistemes encastats en molts dels dispositis que usem a diari, com poden ser routers adsl, càmeres fotogràfiques digitals o fins i tot dins del nostre vehicle.

Estudi de mercat

Sistemes encastats

Actualment al mercat podem trobar una àmplia varietat de sistemes encastats a baix preu. Els darrers anys aquest tipus de productes han adquirit una gran popularitat sobretot entre aficionats cosa que ha fet que sortissin noves iniciatives molt interessants a preus molt competius i disponibles per al públic en general.

Entre tot el ventall de productes que es poden trobar al mercat comentarem els productes *'Arduino Leonardo'*, *'NXP LPC1769'* i la *Raspberry PI*, com a mostra de tres productes força coneguts i de reconegut èxit.

Arduino Leonardo

Arduino es una plataforma de maquinari obert basat en una placa amb un microprocessador i un entorn de desenvolupament. La versió 'leonardo' és un dels molts models de plaques Arduino que existeixen, es basa en un processador ATmega32u4 de 16Mhz, disposa d'una memòria flash de 32Kb i de 2.5Kb de memòria SRAM.

La placa disposa de 20 sortides 7 de les quals poden actuar com a ADC. Alhora també es pot disposar de connectivitat SPI a través del port ICSP.

Es ideal per a principiants, disposa d'un entorn de desenvolupament molt complet. Disposada d'unes llibreries fàcils d'utilitzar molt ben documentades i es fàcil trobar un munt d'exemples per internet.

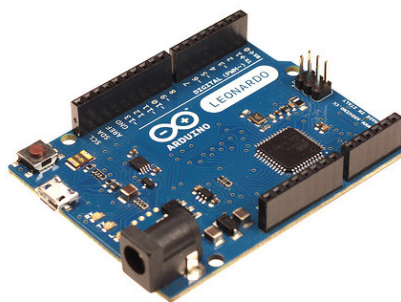


Figura 7 - Placa Arduino Leonardo

Microprocessador	ATmega32u4
Velocitat del rellotge	16Mhz
Arquitectura del processador	8 bits
Voltatge de funcionament	5V
Voltatge d'entrada	7-12V
Memoria Flash	32 KB (ATmega32u4)
SRAM	2.5 KB (ATmega32u4)
Pins E/S Digitals	20
Conectivitat a baix nivell	ADC,SPI,I2C
Baix consum	Idle, ADC reduction,Power Save,Power down,stanby, Extended stanby.
Ethernet	---
Relloige de Temps Real	---
Preu	20€

LPC 1769

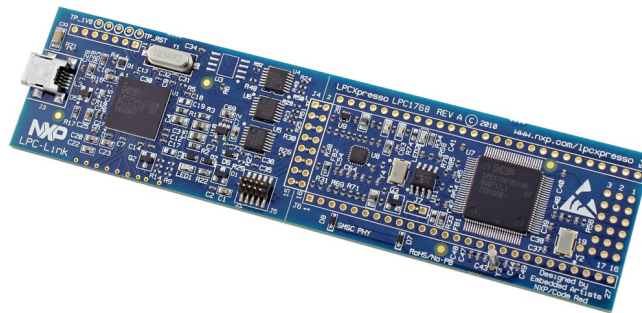


Figura 8 - Placa LPC1769 de NXP

La placa LPC 1769, es una versàtil sistema encastat que disposa d'un processador Cortex-M3 de fins a 120Mhz, 512Kb de memòria flash i 62Kb de memòria SRAM.

La seva principal avantatge és que disposa de moltes entrades i sortides, de rellotge en temps real i de modes de baix consum.

El fabricant proporciona també un entorn de desenvolupament basat en Eclipse amb moltes opcions entre les que destaquen el fet de poder *debugear* el codi directament sobre la placa.

Hi ha documentació i suport per part del fabricant, i algunes llibreries amb exemples que no acaben d'estar del tot ben indexats i explicats. La documentació només esta disponible en anglès. Tot això fa que es requereixi una certa experiència per a treballar amb ella.

Microprocessador	Cortex -M3
Velocitat del rellotge	120Mhz
Arquitectura del processador	32 bits.
Voltatge de funcionament	5V
Voltatge d'entrada	4.5-14V
Memoria Flash	512Kb
SRAM	64Kb
Pins E/S Digitals	70
Conectivitat a baix nivell	ADC,DAC,I2C,SPI,PWM
Baix consum	Sleep Mode, Deep sleep,power down, deep power down
Ethernet	Si
Rellotge de Temps Real	Si

Preu	20€
------	-----

Raspberry pi – Model B

És pot considerar un ordinador complet ja que disposa d'entrada per a teclat i ratolí, sortida per a monitor i executa GNU/Linux. Es va crear per a estimular l'estudi de les ciències de de la computació a les escoles i té un preu molt interessant per les característiques incloses. En quan a connectivitat disposa de 8 pins GPIO amb suport per a I2C, SPI i UART.

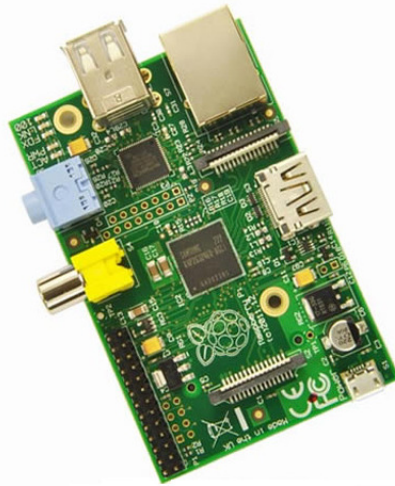


Figura 9 - Placa Raspberry pi

Microprocessador	ARM1176JZF-S ARM-11
Velocitat del rellotge	700Mhz
Arquitectura del processador	32 bits RISC
Voltatge de funcionament	5V
Voltatge d'entrada	4.5-14V
Memoria Flash	512Kb
SRAM	512Mb
Pins E/S Digitals	8
Connectivitat a baix nivell	8 x GPIO, SPI, I ² C, UART
USB	12
Ethernet	Si
Rellotge de Temps Real	---
Preu	40€

Artur Nebot Garrigó

Comunicació entre els dispositius

Per a la comunicació entre els dispositius hi havia una gran varietat de tecnologies a escollir. Encara que dins els requeriments del client, s'especificava la tecnologia i el dispositiu a utilitzar, compararem aquesta tecnologia amb una altra que també hagués resultat completament vàlida per al desenvolupament del projecte.

L'elecció d'una tecnologia o una altra depèn d'alguns factors com els requeriments o necessitats i el preu en la seva implementació

Si el projecte requerís la instal·lació del sistema captador de dades en una zona de difícil accés com per exemple a l'alt d'una muntanya, sempre i quan hi hagués cobertura GSM l'elecció del sistema GPRS resultaria irrefutable.

Sense fils 2.4Ghz (Wifi)

Es un mecanisme de connexió entre dispositius electrònics sense fils format per un conjunt d'estendards oberts. Treballa a una freqüència de 2.4Ghz encara que hi va versions a 5Ghz i a 7.2Ghz.

Els dispositius es connecten a través d'un punt d'accés sense fils.

Entre les principals característiques es troba :

- Baix cost
- Limitació de la distància entre captador i receptor
- Requereix la instal·lació d'una xarxa sense fils
- Freqüència d'ús lliure
- Protocol estàndard

GSPR / 3G

Forma part de la tercera generació de transmissió de dades i veu a través de la telefonia mòbil mitjançant UMTS.

Els serveis associats al 3G proporcionen la possibilitat de transferir tan veu com dades.

- No hi ha limitació de distància entre captador i receptor
- Cal que hi hagi cobertura.
- Cost més elevat comparat amb el wifi
- Requereix la subscripció a un operador



Figura 10 - Mòdul 3G/GPRS compatible amb Arduino i Raspberry pi

Descripció funcional

Sistema total

El conjunt del sistema esta format per un mínim de dos nodes, enllaçats mitjançant una connexió sense fils.

El primer node s'encarrega de capturar les dades, té instal·lats uns sensors que li permeten capturar una sèrie de magnituds físiques i un mòdul de xarxa que li permet comunicar-se amb l'exterior.

El node capturador de dades, al iniciar-se farà una primera comunicació amb el centre de receptor de dades per tal de descarregar la configuració dels intervals i configurar el rellotge. Un cop superat amb èxit aquesta primera comunicació començarà a informar dels valors capturats segons els períodes de temps configurats.

Es pot veure una descripció detallada de la mota al següent apartat ['Mota'](#)

El segon node, executa una aplicació web amb una interfície que permet consumir les dades reportades per el node capturador de dades. Alhora també disposa d'una funció per poder configurar els intervals entre lectures i enviaments de dades i una plana web per mostrar les dades rebudes.

Es pot veure una descripció detallada de tot els processos funcional al apart [Centre de recepció de dades](#)

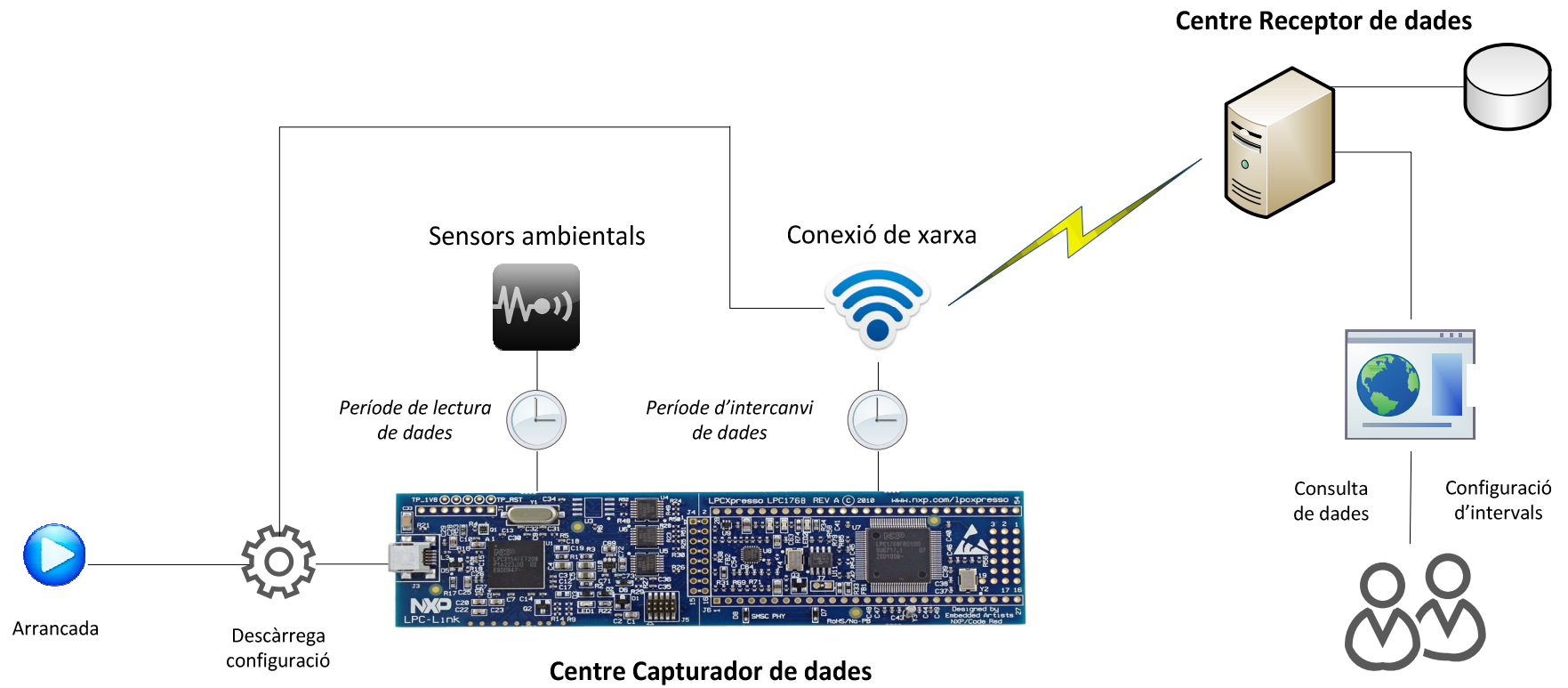


Figura 11 - Esquema general del funcionament del sistema.

A la figura 11 es pot veure el funcionament general del sistema.

Mota

La mota és el node principal del nostre sistema i fa les funcions de captura de dades dels sensors, transformació dels valors llegits a dades, i enviament de les dades capturades cap al 'Centre receptor de dades'.

La mota té dos modes de configuració, funcionament que s'especifica a través del fitxer `app_config.h`, un primer mode anomenat configuració local i un altre anomenat configuració remota.

Amb la configuració local els intervals i el rellotge s'especifiquen a través d'aquest fitxer i no es poden canviar, si la configuració és remota, els intervals de temps entre lectures i enviament, i la sincronització del rellotge s'obtindran del servidor.

Quan la mota arranca, llegeix la configuració del fitxer `app_config.h`, si es troba activat el valor configuració externa, farà una primera connexió amb el 'centre receptor de dades' per a sincronitzar el rellotge i descarregar els intervals entre lectures i enviaments.

El següent diagrama de la figura 12, mostra l'esquema funcional de la mota, els processos marcats en negreta es detallen sota l'esquema.

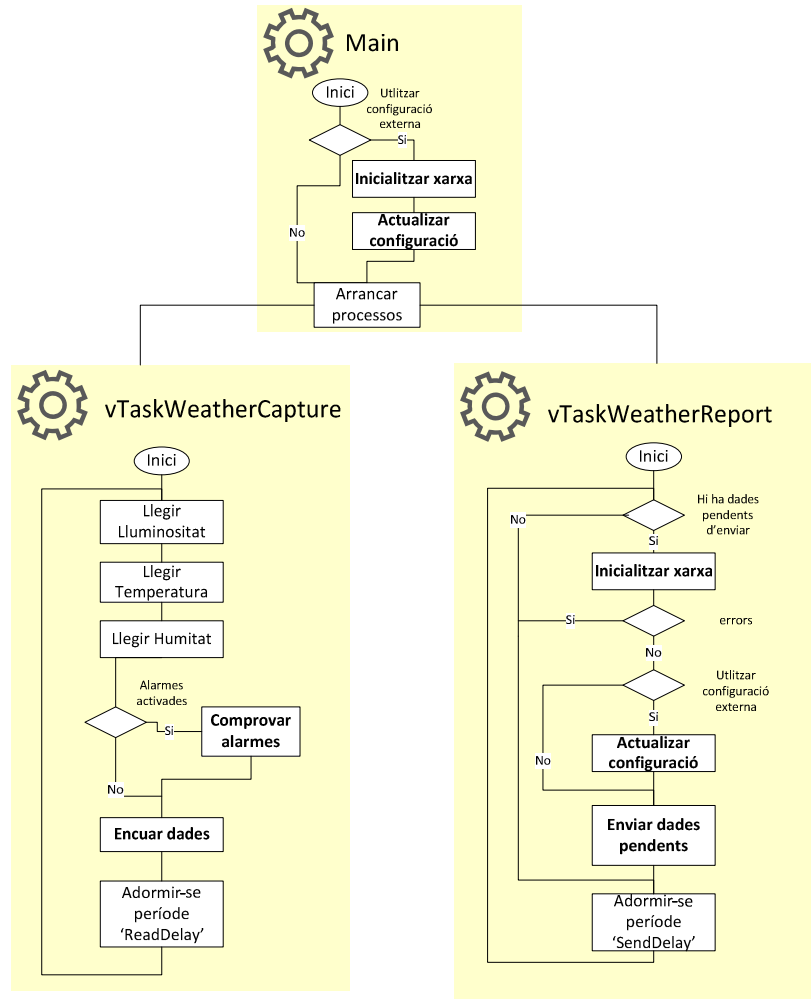


Figura 12 - , descripció funcional de la lògica de la mota.
Les tasques marcades en negreta es detallen a continuació.

Inicialització de la xarxa

El procés d'inicialització de xarxa, ens permet deixar el dispositiu llest per a poder iniciar l'intercanvi de dades amb el centre receptor de dades.

Inicialment inicialitzarà el dispositiu físic WiFly per a poder realitzar connexions de xarxa, i un cop llest, configura la xarxa per tal que el dispositiu quedi preparat per al proper intercanvi de dades.

S'habilita també una sèrie de comprovacions d'estat i de resposta de la xarxa per tal d'assegurar-se el correcte funcionament de tot el procés, la correcta inicialització del dispositiu i la correcta configuració de la xarxa.

En cas de que es produeixi algun error dins d'aquest procés, el sistema sabrà com respondre, repetint el procés fins a un màxim d'intens o esperant un temps de carència per a tornar-ho a intentar.

A la figura 13 es mostra la lògica del procés d'inicialització del dispositiu i la connexió a la xarxa.

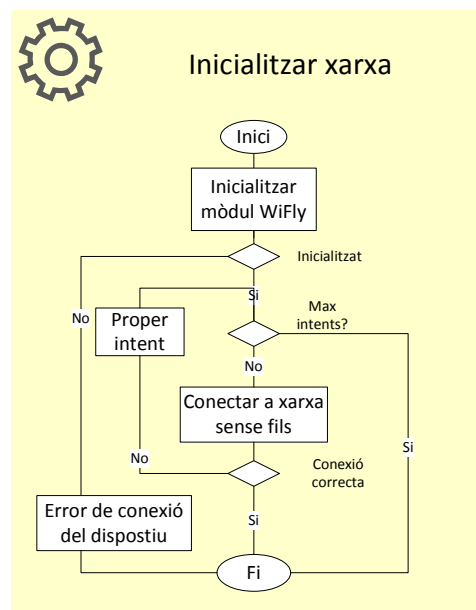


Figura 13 -Procés d'inicialització i connexió de la xarxa.

Actualitzar configuració

La configuració del dispositiu es una tasca que res produeix reiteradament durant el funcionament habitual del sistema.

Aquesta es realitza durant el procés d'arrencada de la mota, i s'aprofita la tasca d'enviament de les dades per mirar si hi ha novetats a la configuració per tal d'aplicar-les sense la necessitat de tornar a iniciar el sistema captador.

Té implementats un conjunt de mecanismes per a protegir-se en cas de fallada, en cas de no poder aplicar la nova configuració mantindrà la última configuració bona coneguda.

A la figura 14 es mostra la lògica del procés d'obtenció de la configuració i actualització del dispositiu

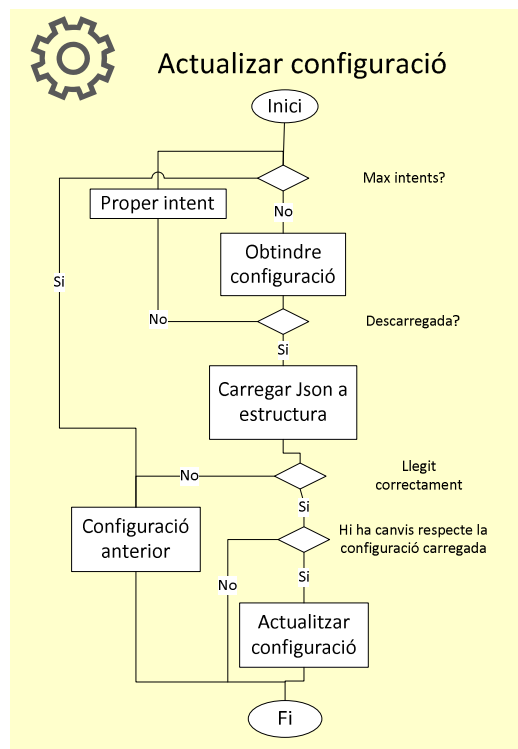


Figura 14 - procés d'actualització de la configuració del sistema captador de dades

Enviament

El procés d'enviament de dades per part del sistema captador de dades és una de les tasques primordials per al correcte funcionament del sistema.

Aquest procés s'executa cada cert temps i envia les dades que es troben en estat pendent d'enviament en el moment de l'execució.

Resulta important intentar minimitzar el grau de fallida i tolerar fins a un cert grau les fallades que puguin produir-se en el moment de l'enviament, per això si un dada no es pot enviar es fa el possible per tal que no es perdi, intentar tornar-la a enviar fins a un màxim d'intents prèvia espera d'un període de carència, i si no s'aconsegueix, emmagatzemar-la per a poder-la enviar en un moment en el que la connexió es trobi disponible,

Dins l'apartat [Mida del buffer de dades](#) es fa un estudi de la capacitat d'emmagatzematge del buffer, és a dir els número de captures que es pot emmagatzemar en cas de fallada abans de començar a perdre dades.

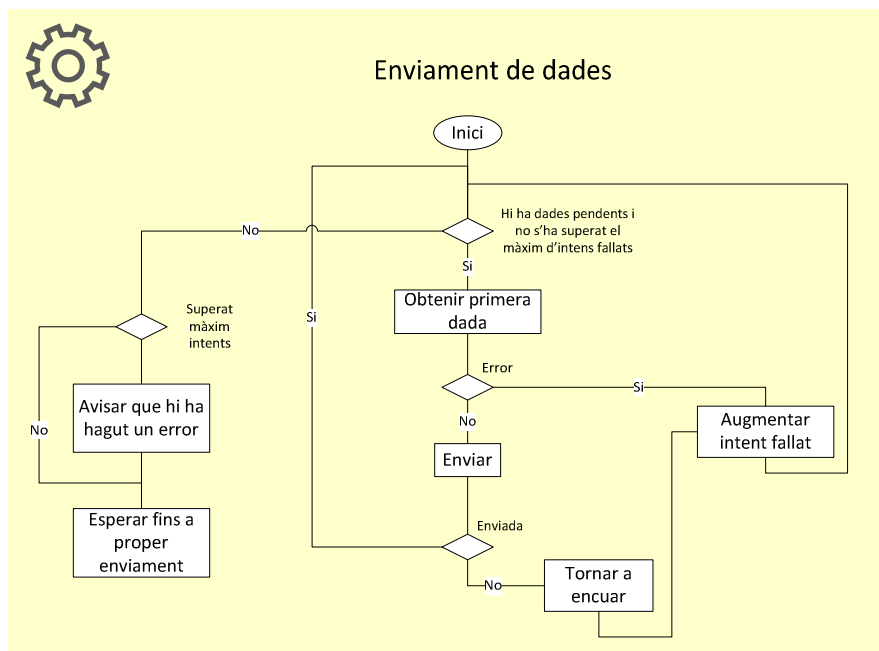


Figura 15 - procés d'enviament de les dades

Comprovació de les alarmes

Una funcionalitat addicional implementada dins el 'sistema captador de dades' és la possibilitat de configurar una alarma per cada sensor quan aquest llegeixi un valor fora d'un rang fitat i definit com a esperat.

En cas que es produeixi un valor fora de rang, per exemple una temperatura superior al màxim esperat s'activarà l'alarma per aquest sensor.

A la figura següent es mostra la lògica de funcionament del sistema que fa saltar les alarmes.

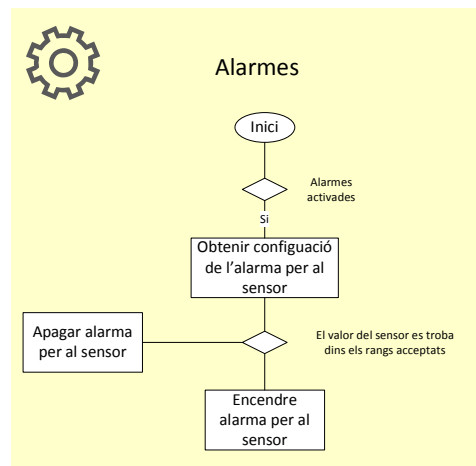


Figura 16 - procés de comprovació de les alarmes

Centre de recepció de dades

El sistema captador de dades no té sentit sense l'existència d'un sistema que permeti obtenir les dades, emmagatzemar-les i poder tractar-les. Aquest requisit és el que justifica la creació d'un 'centre de recepció de dades'.

El centre de recepció de dades esta compost per un servidor web i un aplicatiu web, aquest fa les funcions

- Configuració de la mota (en cas de configuració remota configurada)
- Recepció de les dades
- Emmagatzemament de les dades
- Mostrar les dades.

Aquest es troba instal·lat en un segon ordinador i connectat amb la mota mitjançant una connexió sense fils.

Configuració de la mota

Si la mota te configurat el paràmetre `USE_EXTERNAL_CONFIG` dins els fitxer `app_config.h`, al arrancar es connectarà al centre de recepció de dades per a obtenir la seva configuració.

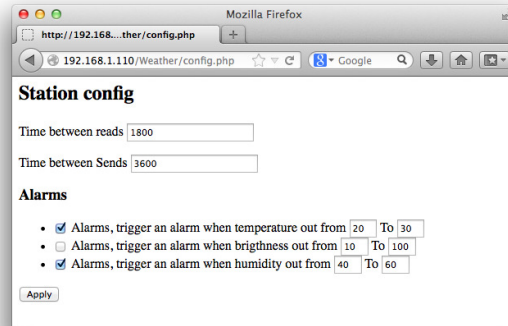


Figura 17 - Plana de configuració de la mota

La figura 17 mostra la plana web de configuració del a mota disponible a la URL `http://receptor/config.php` que proporciona una interfície d'usuari bàsica per a dur a terme la configuració de les alarmes i els intervals de lectura i enviament de dades

Les dades introduïdes mitjançant aquesta pantalla seran consumides per la mota per configurar els seus paràmetres, a la figura 18 veiem la interfície mitjançant la qual la mota obté les dades de la seva configuració.



Figura 18 - Mostra de la interfície per a l'obtenció de la configuració

Mostra de les dades capturades

Es important que l'usuari pugui consultar les dades capturades per la mota a través d'una interfície d'usuari simple i còmoda.

A la figura 19 es mostra la pantalla de visualització de les dades, que es va actualitzant cada cert període de temps per a mostrar les novetats rebudes per els sensors.

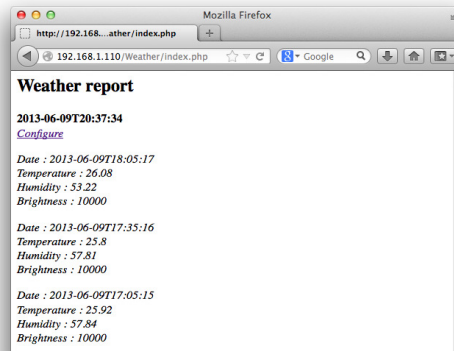


Figura 19 - Mostra de la plana de visualització de dades

Alimentació de les dades

Per tal que el sistema receptor de dades pugui consumir les dades generades per el centre captador de dades, caldrà una interfície web que proporcioni un mètode de inserció de dades.

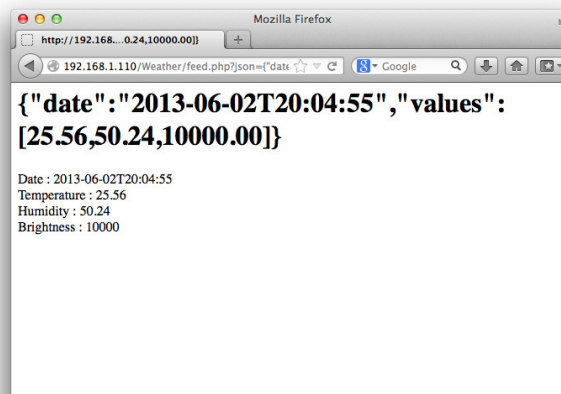


Figura 20 - Mostra de la interfície d'alimentació de dades del sistema receptor de dades

Descripció detallada

Dins el punt anterior '[Descripció funcional](#)' s'ha explicat el funcionament del sistema i la seva lògica, sobretot a nivell de programari. En aquest apartat entrarem una mica més en detalls tècnics sobre la seva implementació que mancaven pendents.

Estructura de dades i missatges.

Per a poder assolir amb èxit els objectius del nostre projecte és important emmagatzemar les dades capturades dins unes estructures adequades, i implementar un protocol d'intercanvi de dades que sigui capaç de mantenir la comunicació entre tots els nodes afectats.

La informació que requerim emmagatzemar per al correcte funcionament del sistema es

- Dades de configuració
- Dades capturades per els sensors

Dades de configuració

Dins les dades de configuració entren tots els paràmetres configurables per part de l'usuari, en el nostre cas són

- Interval entre lectures
- Interval entre enviaments
- Alarmes (per a cada alarma).
 - Estat de la alarma (activat o desactivat)
 - Valor màxim acceptat
 - Valor mínim acceptat

L'estructura utilitzada per assolir aquest objectiu és la següent :

Per als intervals s'utilitzen dues variables globals a tot el sistema, de tipus enter definides al fitxer `app_config.c`.

Artur Nebot Garrigó

```
uint32_t READ_DELAY_SEC = 15*60;  
uint32_t SEND_DELAY_SEC = 30*60; //120;
```

Si no es modifiquen o no es configuren des del centre tenen un valor per defecte de 15 minuts per a les lectures i 30 minuts per als enviaments.

Per a les alarmes s'ha creat un vector d'estructures indexat per el identificador del sensor definit a `captured_data.h`. Cada estructura suporta les tres propietats esmentades anteriorment.

```
enum {HUMIDITY = 0,TEMPERATURE = 1,BRIGHTNESS= 2} SensorType;  
  
struct AlarmDefinition {  
    bool Enabled;  
    double MinAcceptableValue;  
    double MaxAcceptableValue;  
} m_alarms[MAX_SENSORS];
```

El missatge entre el centre receptor de dades i el centre captador de dades per a la transmissió de les dades de configuració és el següent :

```
{ "CurrentDate": "2013-06-09T11:11:40"
  , "ReadDelay": "1800"
  , "SendDelay": "3600"
  , "Alarms": [ { "Enabled": 0           // Sensor temperatura
                , "MinValue": 20
                , "MaxValue": 25
                }
              , { "Enabled": 0           // Sensor humitat
                , "MinValue": 50
                , "MaxValue": 60
                }
              , { "Enabled": 0           // Sensor lluminositat
                , "MinValue": 1000
                , "MaxValue": 10000
                }
            ]
}
```

Per a la transformació del missatge JSON a l'estructura de dades escrita en C ens em ajudat de la llibreria [jsmn](#).

Dades capturades per als sensors

Per a les dades capturades necessitem saber la data de la captura i els valors obtinguts. Per tal de no sobrecarregar el procés s'ha mantingut l'estructura el més simple possible.

```
enum {HUMIDITY = 0,TEMPERATURE = 1,BRIGHTNESS= 2} SensorType;

struct CapturedData {
    char date[19];
    float value[MAX_SENSORS];
};
```

CapturedData.date es un string que conté la data de la captura en format YYYY-MM-DDTHH:mm:ss

CapturedData.value[] es un vector indexat per el tipus de sensor segons el enumerat SensorType

Artur Nebot Garrigó

Missatge d'intercanvi de dades

```
{ "date": "2013-06-09T05:31:53"
  , "values": [ 59.35           // Sensor humitat
                , 25.38         // Sensor temperatura
                , 1000          // Sensor lluminositat
              ]
}
```

Per a la transformació del missatge JSON a l'estructura de dades escrita en C ens em ajudat de la Llibreria [jsmn](#).

Mida del buffer de dades

El procés principal, explicat al punt '[mota](#)' de l'apartat [descripció funcional](#), explica com es generen i consumeixen les dades. S'explica l'existència d'una cua on cada procés, el generador i el consumidor *encua* i *desencua* les dades.

L'existència d'un buffer per emmagatzemar les dades implica que cal especificar una mida per aquest ja que la memòria del nostre sistema es limitada.

La mida del buffer d'emmagatzemament es troba definida per la constant `QUEUE_DATA_SENSOR_MAX_ITEMS` i en cas de voler modificar-lo caldrà fer-ho abans de la compilació

Això vol dir que podem emmagatzemar fins a `QUEUE_DATA_SENSOR_MAX_ITEMS` dades en la memòria de captures pendents d'enviar abans de perdre les noves captures.

Per tant cal assegurar-se que generem menys de `QUEUE_DATA_SENSOR_MAX_ITEMS` dades entre la generació i l'enviament de dades caldrà calcular.

$$\frac{\text{dades generades}}{\text{temps captura dades}} \cdot \text{temps enviament} < \text{QUEUE_DATA_SENSOR_MAX_ITEMS}$$

Sensors

Sensor de lluminositat

Per tal de llegir la lluminositat de l'entorn al sistema captador de dades hi ha connectada una cèl·lula fotovoltaica analògica FW-135 llegida a través del convertidor ADC (*Analog to digital*) incorporat a la placa.

Per a connectar el sensor connectarem l'entrada de la cèl·lula fotovoltaica directament a la font d'alimentació i la sortida a una resistència pull-down. El punt entre la resistència fixa final pull-down i el nostre sensor (que actua com una resistència variable) el connectarem a la mota.

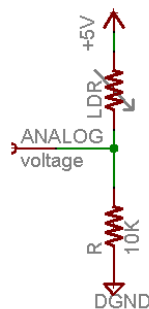


Figura 21 - Esquema de connexió del sensor font: learn.adafruit.com

Això modificarà el corrent que flueix entre les dues resistències segons el nivell de lluminositat, fitat en un màxim i un mínim segons les dues resistències.

Conversió de lectura del sensor a lúmens.

Mitjançant la lectura del ADC llegirem el voltatge capturat en aquest moment segons el nivell de resistència oferta per el sensor, aquest valor realment no es entenedor i per a que resulti útil cal convertir-lo a lúmens.

Segons el nivell de lluminositat que vulguem mesurar incorporarem una resistència pull-down de 10K o 100K. La primera la utilitzarem per a entorns amb poca llum (per exemple una habitació) i la segona en cas d'instal·lar l'estació a l'exterior.

Artur Nebot Garrigó

El nostre sistema suporta els dos tipus de resistència però caldrà especificar quin tipus de resistència volem utilitzar, per això fins el fixter app_config.h descomentarem la línia apropiada segons la resistència incorporada.

```
#define LUM_RESISTOR          RESISTOR_10K      // For INDOOR
#define LUM_RESISTOR          RESISTOR_1K      // For OUTDOOR
```

La relació entre el voltatge i la lluminositat no es lineal, sinó que segueix una evolució logarítmica. Aquesta conversió es complexa i s'acostuma a utilitzar aproximacions

Per tal de facilitar els càlculs i degut a la poca precisió de la lectura del sensor ens em ajudat d'unes taules de conversió per a la conversió a lúmens

Es pot consultar les taules utilitzades a l'annex "[Taules de conversió de voltatge a lúmens](#)"

Sensor de temperatura i humitat

S'ha instal·lat el sensor de temperatura SHT11 fabricat per Sensirion amb el mòdul de presentació parallax, per tal de facilitar la connectivitat.

Es tracta d'un sensor amb capacitat per a mesurar la temperatura ambiental i la humitat relativa, amb una precisió de mig grau i un consum de 80uW (a 12 bits , 3V i 1 mesura per segon). La connectivitat el sensor es a través d'una interfície 2-wire.

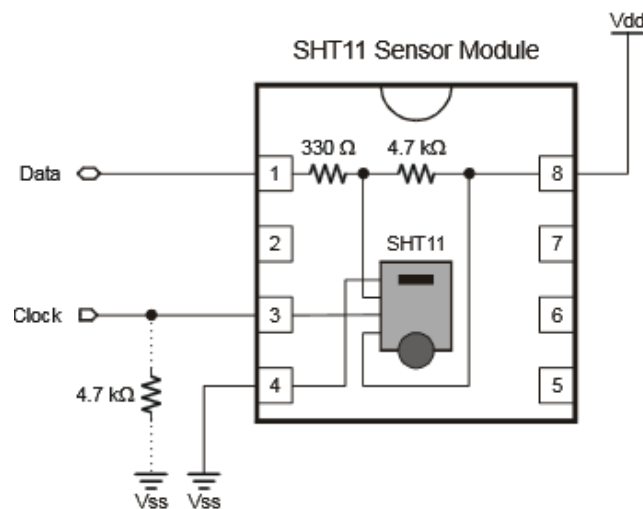


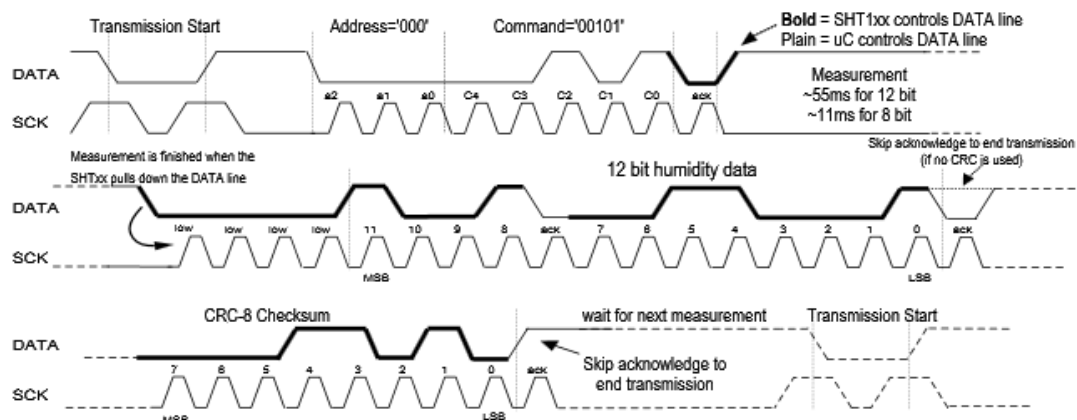
Figura 22 -,Esquema de connectivitat del sensor

Al tractar-se d'un sensor digital no cal utilitzar fórmules de conversió del voltatge al valor real, tot i això ens cal realitzar una sèrie d'operacions per a poder transformar el valor llegit en la temperatura i la humitat real.

Dins els annexos a l'apartat [Datashet del sensor Sensirion SHT11](#) es troba l'enllaç al *datasheet* del sensor on s'especifica les operacions a realitzar.

Per tal de connectar el sensor s'ha utilitat una interfície 2-wire, tot i això el protocol que implementa el sensor no es compatible amb l'estàndard I²C, per aconseguir comunicar-se amb el sensor ha calgut implementar el protocol específic del sensor.

A la figura 23 es troba el procés de inicialització del dispositiu, enviament de la comanda (lectura d'humitat o temperatura) i lectura del valor obtingut.



23 -.Esquema de connectivitat del sensor estreta del datasheet proporcionat per el fabricant.

Figura

Baix consum

La implementació del baix consum s'ha implementat al sistema captador de dades, per tal de permetre una gestió òptima de l'energia i amb la intenció que pugui ser alimentant mitjançant una bateria.

La placa LPC1769 sobre la que es troba muntat el sistema captador de dades suporta diferents modes de baix consum, cada un dels modes presenta unes característiques determinades i un nivell de consum diferent.

Com més baix és el nivell de consum més dispositius interns de la placa es desconnecten, i per tant a l'hora de despertar-lo caldrà tenir-ho en especial consideració.

Un altre factor també molt important és el temps que tarda cada nivell en canviar del mode adormir al mode plenament operatiu.

- **Sleep mode**

El rellotge de la placa es troba desconnectat, tot i això el sistema d'interrupcions es troba funcionant. Qualsevol interrupció habilitada podrà despertar el sistema.

Un cop executada la rutina d'atenció a la interrupció es pot escollir si tornar a adormir-lo o continuar amb l'execució normal mitjançant un registre de configuració.

El temps entre el canvi d'estat de 'sleep mode' a funcional resulta molt ràpid.

- **Deep sleep mode**

Paregut al mode 'sleep mode' amb la diferència que la memòria flash es troba en mode stand by. El consum es redueix fins a un valor molt baix.

- **Power down mode**

Implementa les mateixes funcionalitats que el mode deep sleep mode amb la diferència que la memòria flash es troba completament aturada.

Les interrupcions, el oscil·lador principal i tots els rellotges es troben completament aturats.

L'únic rellotge que es troba operatiu es el rellotge en temps real, les interrupcions d'aquest rellotge poden despertar el dispositiu.

Caldrà esperar a que la memòria es trobi preparada per tenir la placa en un mode funcional.

- **Deep power down mode**

El chip es troba completament aturat. El procés de despertar en Deep power-down mode només es produirà quan es rebi un senyal de reset extern o el rellotge en temps real produeixi una interrupció.

Implementació del baix consum al sistema captador

En el nostre projecte el sistema de baix consum implementat ha estat el nivell '*sleep mode*', cosa que ens permet mantenir la simplicitat en la implementació i despertar-lo ràpidament, permetent temps de entre captura de dades o entre enviaments molt reduïts.

Com s'ha realitzat la implementació?

Per implementar el mode baix consum s'ha aprofitat la tasca `void vApplicationIdleHook(void)`. Aquesta s'executa en el moment en que el planificador de processos del sistema operatiu FreeRTOS no té cap tasca en estat preparat per a la seva execució, resulta doncs un moment ideal per a modificar l'estat de la placa cap a un estat en mode de baix consum.

Per a comprovar que realment el consum de la placa baixa ens em ajudat del multímetre, connectant-lo en mode Amperímetre a la alimentació de la placa i tancant el circuit.

Altres apartats

Joc de proves

Testeig dels sensors

Per tal de comprovar el correcte funcionament del sensor SHT11 i corroborar que les lectures produïdes eren correctes, s'ha connectat el sensor a una placa Raspberry Pi i utilitzant un controlador ja desenvolupat per un tercer s'ha comprovat que el nostre sistema produeix les mateixes lectures.

Ens em ajudat d'aquest article per a obtenir el codi del sensor i realitzar al compilació i posterior prova del codi.

<http://www.john.geek.nz/2012/08/reading-data-from-a-sensirion-sht1x-with-a-raspberry-pi/>

Proves de connexió

S'ha realitzat un a sèrie de proves de connectivitat entre el sistema captador de dades i el centre receptor dades per comprovar el comportament del sistema en casos en que es produeixin caigudes.

Les caigudes de xarxa son factors externs que no es poden controlar, el nostre sistema haurà de ser capaç de detectar el problema i tornar a intentar la connexió passat un període de carència, o en el pitjor dels cassos mostrar un missatge d'error i aturar-se.

Connectar una xarxa no disponible

El sistema intentarà al arrancar la connexió de xarxa fins a tres vegades , en cas que no pugui establir la connexió s'aturarà i mostrarà el missatge "*ERROR No network! System shutdown*"

Artur Nebot Garrigó

Desconnectar la xarxa sense fils durant el funcionament del sistema.

El sistema intentarà enviar les dades, si per alguna raó no pot arribar a enviar-les, emmagatzemarà les dades per a realitzar un enviament posterior.

S'ha comprovat que si el període de temps es superior al màxim suportat per la capacitat d'emmagatzemament de les dades, aparti de la següent dada generada ja no es podrà emmagatzemar, mostrant el error. *"Warning! Full Buffer, captured data can't be saved!"*

Ampliacions possibles

Hi ha algunes opcions que m'hagués agradat implementar i trobo que afegirien un valor al producte força interessant.

Afegir-les no ha estat possible degut a la curta durada del projecte, però és important que en quedi constància per tal d'implementar-les en futures versions.

- Exportar a CSV les dades reportes
Caldrà modificar el centre receptor de dades per a què generi un fitxer separat per coma de les dades emmagatzemades.
Es una opció interessant ja que permet la exportació ràpida i fàcil a molts altres sistemes com per exemple una fulla de càlcul.
- Poder enviar les dades en un bloc paginat.
És un procés interessant en el cas que el sistema captador generi moltes dades i els temps entre enviaments siguin relativament llargs, per tal d'evitar que l'enviament de moltes dades d'una en una poder-les enviar en bloc.
- Reportar les alarmes al sistema receptor de dades
Afegir un missatge nou que notifiqui al sistema receptor de dades les alarmes produïdes per el sistema captador de dades.
Una altra opció per implementar el mateix comportament és comprovar les dades al sistema receptor, amb l'inconvenient que es perdria el factor dels temps real, la data de captura no té per què se la mateixa que la de l'enviament.
- Incorporació d'un sensor de pressió atmosfèrica
M'hagués agradat incorporar un sensor de pressió atmosfèrica per tal de poder captura també aquesta dada.
La incorporació en un futur no resulta complicada ja que l'estructura de dades és fàcilment escalable.

Viabilitat tècnica

En aquest apartat estudiarem la viabilitat tècnica del projecte desenvolupat, i esmentarem els punts i decisions preses en el disseny i implementació per tal d'assolir els objectius i poder fer viable el projecte.

Els punts comentats es basen en el llistat [d'objectius del projecte](#) explicat a la introducció del document.

Protocol d'intercanvi de dades obert.

L'ús d'un protocol d'intercanvi de dades obert facilita l'escalabilitat del nostre projecte, facilitant la integració de la nostra solució amb d'altres ja implementats i proporcionant un augment de les funcionalitats i característiques del sistema.

Es amb aquest objectiu per el que s'ha vetllat en definir un protocol d'intercanvi de dades comprensible i fàcilment llegible per altres aplicatius externs.

Per tal d'assolir aquest objectiu s'ha escollit el protocol d'intercanvi de dades JSON, els motius principals de la seva elecció han estat

Molt lleuger

Les característiques tècniques de la mota fan que disposem d'una memòria i capacitat de procés limitada,

El llenguatge JSON proporciona un protocol d'intercanvi de dades molt lleuger degut a la pròpia naturalesa del format, que redueix al mínim la definició de l'estructura de les dades prioritzant el valor de les dades. Els paquets ocupen molt poc espai.

Suport per a multitud de llenguatges de programació

Es troba suportat de forma directa o mitjançant llibreries externes per una àmplia varietat de llenguatges de programació, augmentant les possibilitats de comunicació del nostre sistema amb l'exterior.

Entre els principals llenguatges suportats es troben *ActionScript*, *C*, *C++*, *C#*, *ColdFusion*, *Common Lisp*, *Delphi*, *E*, *Eiffel*, *Java*, *JavaScript*, *ML*, *Objective-C*, *Objective CAML*, *Perl*, *PHP*, *Python*, *Rebol*, *Ruby*, *Lua* y *Visual FoxPro*.

Àmpliament estès

Actualment s'utilitza en un munt d'aplicacions, sobretot en l'àmbit dels serveis web i Internet. Grans corporacions com Google o Yahoo l'implementen habitualment en el seu programari.

Transparència entre la implementació entre el captador i el receptor.

Cal assegurar-se que canvis en la implementació interna d'una part no afectin a la resta de mòduls. Per assolir aquest objectiu s'ha definit una interfície de comunicació entre el sistema captador i el sistema receptor, i una estructura de dades per als intercanvis d'informació.

config.php?Get

Obté la informació de configuració mitjançant una estructura JSON. El detall de l'estructura es pot consultar a l'apartat [Estructura de dades i missatges](#).

feed.php?json=

Alimenta el receptor de dades mitjançant una estructura JSON. El detall de la definició de l'estructura es pot consultar a l'apartat [Estructura de dades i missatges](#).

Capturar les magnituds físiques

Dins la llista d'objectius s'especifica un conjunt de magnituds físiques de l'atmosfera que caldrà mesurar per tal d'assolir els requisits esperats del projecte.

Per acomplir-los el sistema captador de dades incorpora els següents dispositius :

Sensor Sensirion SHT11

Cèl·lula fotovoltaica FW-135

S'ha implementat una llibreria per permetre la connectivitat i la lectura de les dades dels sensors.

Grau d'autonomia

Per tal de proporcionar un alt grau d'autonomia el sistema captador de dades es trobarà en repòs mentre no executi ni les tasques de lectura dels sensors ni les tasques d'intercanvi d'informació.

El detall en la implementació del sistema de reducció de l'energia s'explica a l'apartat [Baix consum](#)

Tolerància a fallades.

Per tal de proporcionar un grau de tolerància de dades, i fitar els casos on es poden produir pèrdua de dades s'ha implementa una sèrie de mecanismes de seguretat i control sobre l'enviament de les dades.

Els mecanismes per a donar garantia i controlar els casos en els que es produeixen fallades, es troben explicats dins l'apartat ['Mota'](#)

Les proves realitzades per a testejar el comportament en cas de fallada es troben explicats a l'apartat [Proves de connexió](#)

Valoració econòmica

No únicament les raons tècniques i l'assoliment dels objectius estableixen la viabilitat o l'èxit d'un projecte, dins el mercat empresarial un dels principals indicadors de l'èxit d'un projecte és els recursos econòmics invertits en el seu desenvolupament

Recursos Materials :

Inclou únicament els dispositius instal·lats a la placa.

LPC1769	20€
Parallax SHT11	41,43 €
Fotocèlula FW150	1,58€
Mòdul WiFly	26,4 €
CP2102	3,7 €
<i>Llicències de programari</i>	0
Sistema consumidor de dades.	93,11€
HP Proliant Microserver	213,55€
Tp-Link TL-WN722N	12,77
<i>Llicències de programari</i>	0
Sistema receptor de dades	226.32
Total	319,43

Altres recursos materials utilitzats per al desenvolupament

Artur Nebot Garrigó

Altres materials utilitzats per al desenvolupament però que no formen part del paquet del producte.

Cablematic - Soldador eléctrico de 30W	10,69 €
Hama - Digital Multimeter "EM393"	14,56 €
Protoboard 830 contactos Breadboard placa prototipos sin soldadura	6,49 €
Total	31,74€

Els ordinadors utilitzats per al desenvolupament no s'inclouen dins la valoració econòmica ja que la no són d'ús exclusiu per aquest projecte i ja havien sigut adquirits amb anterioritat.

Cost de desenvolupament

Perfil	Quantitat	Preu	Total
Hores de formació	50	10	500
Hores de consultoria	25	60	1500
Hores de desenvolupament	180	42	7560
Hores de documentació	30	20	600
			10160

Valoració econòmica del projecte :

10.511.17 €

Conclusions

Dins d'aquest apartat es descriurà les conclusions personals del projecte per part de l'alumne, utilitzant un llenguatge més personal i formal.

El desenvolupament del projecte ha resultat una tasca força complicada que m'ha portat força mals de caps, sobretot degut a la nul·la experiència prèvia en el tema i al poc temps disponible per a la realització.

Tot i això ha resultat un repte força interessant i m'ha permès aprendre força sobre tema.

La dificultat del tema escollit, i la falta de temps deguda principalment a compaginar el projecte amb la vida laboral, no m'han permès portar el projecte amb el ritme que m'hagués agradat.

Tinc de dir que personalment estic força satisfet amb el producte final, que ha aconseguit funcionar i assolir els objectius marcats, encara que cal reconèixer que al projecte li manca una certa maduresa hi ha alguns mòduls caldria fer-li algunes millores.

Bibliògrafa

- PHP: The Good Parts
<http://shop.oreilly.com/product/9780596804381.do>
- El lenguaje de programación c
<http://www.amazon.es/El-lenguaje-programacion-c-Kernighan/dp/9688802050>
- LPC17xx User manual
http://www.nxp.com/documents/user_manual/UM10360.pdf
- AN10915 Using the LPC1700 power modes
http://www.po-star.com/public/uploads/20120319123122_141.pdf
- FreeRTOS documentation
<http://www.freertos.org/Documentation/code/>
- Adafruit learning system
<http://learn.adafruit.com>
- SHT11 driver for Raspberry pi
<http://www.john.geek.nz/2012/08/reading-data-from-a-sensirion-sht1x-with-a-raspberry-pi/>
- NXP support forums
<http://knowledgebase.nxp.com/>
- Documentació de la llibreria jsnm
<http://zserge.bitbucket.org/jsnm.html>
- Documentació sobre el format d'intercanvi de dades JSON
<http://www.json.org>

Annexos

Taules de conversió de voltatge a lúmens

Font : <http://learn.adafruit.com/photocells/using-a-photocell>

Resistència de 10 K Ω

Ambient light like...	Ambient light (lux)	Photocell resistance (Ω)	LDR + R (Ω)	Current thru LDR +R	Voltage across R
Dim hallway	0.1 lux	600K Ω	610 K Ω	0.008 mA	0.1 V
Moonlit night	1 lux	70 K Ω	80 K Ω	0.07 mA	0.6 V
Dark room	10 lux	10 K Ω	20 K Ω	0.25 mA	2.5 V
Dark overcast day / Bright room	100 lux	1.5 K Ω	11.5 K Ω	0.43 mA	4.3 V
Overcast day	1000 lux	300 Ω	10.03 K Ω	0.5 mA	5

Resistència de 1K Ω

Ambient light like...	Ambient light (lux)	Photocell resistance (?)	LDR + R (?)	Current thru LDR+R	Voltage across R
Moonlit night	1 lux	70 K Ω	71 K Ω	0.07 mA	0.1 V
Dark room	10 lux	10 K Ω	11 K Ω	0.45 mA	0.5 V
Dark overcast day / Bright room	100 lux	1.5 K Ω	2.5 K Ω	2 mA	2.0 V
Overcast day	1000 lux	300 Ω	1.3 K Ω	3.8 mA	3.8 V
Full daylight	10,000 lux	100 Ω	1.1 K Ω	4.5 mA	4.5 V

Datasheet del sensor Sensirion SHT11

<http://www.parallax.com/Portals/0/Downloads/docs/prod/datast/shtx.pdf>

Llibreria JSMN de conversió de dades a JSON

<http://zserge.bitbucket.org/jsmn.html>