
This is the **published version** of the bachelor thesis:

Roca Vázquez, Jordi. *Sistema distribuït de detecció de fuites d'aigua basat en OpenThread i RTOS*. Treball de Final de Grau (Universitat Autònoma de Barcelona), 2026 (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/326581>

under the terms of the  license.

Sistema distribuït de detecció de fuïtes d'aigua basat en OpenThread i RTOS

Jordi Roca Vázquez

9 de febrer de 2026

Resum– En aquest document es presenta la creació d'un prototip domòtic de detecció de fuïtes d'aigua. El sistema està format per tres plaques ESP32 formant una arquitectura distribuïda, una placa col·locada en la zona d'interès amb el sensor, i les dues plaques restants unides per crear un Border Router Thread, element clau del protocol de domòtica Thread per crear xarxes d'interconnexió de sensors i actuadors, i per connectar aquestes xarxes amb xarxes IP, facilitant la gestió dels dispositius. El sistema inclou pantalla, buzzer i protocol de missatgeria MQTT amb l'objectiu de que sigui un prototip versàtil, adaptat tant per cases domotitzades com per les que no. La implementació s'ha fet amb RTOS per garantir un funcionament multitasca eficient i fiable.

Paraules clau– ESP32, FreeRTOS, OpenThread, Thread Border Router, UDP, C, MQTT, IoT, IEEE 802.15.4, Protocol Spinel, sensor de gotes d'aigua, UART, GPIO, SPI, HomeAssistant.

Abstract– This document presents the creation of a home automation prototype for detecting water leaks. The system is composed of three ESP32 boards forming a distributed architecture: one board placed in the area of interest with the sensor, and the remaining two boards paired to create a Border Router Thread, a key element of the Thread smart home protocol for creating sensor and actuator interconnection networks and for connecting these networks to IP networks, facilitating device management. The system includes a display, buzzer, and MQTT messaging protocol with the goal of being a versatile prototype, suitable for both smart homes and non-smart homes. The implementation was done with an RTOS to ensure efficient and reliable multitasking.

Keywords– ESP32, FreeRTOS, OpenThread, Thread Border Router, UDP, C, MQTT, IoT, IEEE 802.15.4, Spinel Protocol, water drop sensor, UART, GPIO, SPI, HomeAssistant.



1 INTRODUCCIÓ - CONTEXT DEL TREBALL

UNA fuïta d'aigua dins d'un habitatge concret és un esdeveniment amb una baixa probabilitat d'ocurrència, però a nivell global és un fenomen que es produeix de manera recurrent, especialment en construccions de mala qualitat o amb una antiguitat ja considerable a causa de l'inevitable desgast dels materials. Aquestes fuïtes poden aparèixer al voltant del vàter, sota aigüeres i electrodomèstics, al sostre, etc., i poden tenir un impacte material i/o econòmic important, especialment si no es detecten i es gestionen ràpidament.

També és important ressaltar que, en molts casos, les fuïtes d'aigua es produeixen sense una interacció prèvia per part de l'usuari, fet que obre la possibilitat de que tinguin lloc en absència de persones a l'habitatge. En aquest context, cobra importància la domòtica, que juntament amb altres avantatges com el confort, accessibilitat o estalvi energètic, permet l'accés remot a l'estat de l'habitatge.

Tenint en compte tot l'esmentat, aquest treball presenta el disseny i implementació d'un sistema domòtic de detecció de fuïtes d'aigua, basat en una arquitectura distribuïda i escalable, fent ús de tecnologies i protocols estandarditzats i comuns en projectes IoT, amb una integració de mecanismes d'alerta tant locals com remots, per garantir la recepció primerenca de l'avís de fuïta en diferents situacions i tipologies d'habitatge.

L'estructura del dossier està composta pel següent: objectius, metodologia, marc teòric, desenvolupament, resultats, conclusions i possible línia de millora.

- E-mail de contacte: Jordi.RocaVa@autonoma.cat
- Menció: Enginyeria de Computadors
- Treball tutoritzat per: Marius Montón Macián (Microelectrònica i Sistemes Electrònics)
- Curs 2025/26

2 OBJECTIUS

Basat en l'esmentat en la secció anterior, he establert els següents objectius, diferenciats en objectiu general, objectius específics i objectius formatius

2.1 Objectiu general

- L'objectiu principal és la creació d'un prototip de sistema encastrat que detecti fuites d'aigua en habitatges i generi alertes locals i remotes.

2.2 Objectius específics

- Dissenyar una arquitectura distribuïda, separant les funcions de detecció i de comunicació a l'usuari.
- Integrar un mòdul de detecció de presència d'aigua.
- Implementar actuadors per tenir una alerta local.
- Establir una xarxa de comunicació local entre els dispositius del prototip.
- Incorporar al sistema connectivitat IP.
- Afegir una comunicació amb una plataforma domètica per tenir alertes remotes en un sistema extern.
- Garantir un funcionament robust i eficient mitjançant l'ús d'un sistema operatiu en temps real i una gestió adequada de tasques.
- Optimitzar recursos del prototip.

2.3 Objectius formatius

- Consolidar coneixements sobre sistemes encastrats, comunicacions sense fils i arquitectures IoT.
- Desenvolupar habilitats en la integració de sistemes domòtics amb plataformes externes i serveis de monitoratge.
- Aplicar i entendre conceptes de sistemes operatius en temps real, com la gestió de tasques, cues i sincronització.
- Adquirir experiència en el desenvolupament d'un projecte complet, des de la definició del problema fins a la implementació i validació del sistema.

3 METODOLOGIA

La metodologia emprada en el desenvolupament del treball és una metodologia àgil híbrida, que combina la metodologia Crystal Clear amb una estratègia de prototipat incremental. Aquesta elecció es basa tant en la naturalesa del projecte com en el context en què s'ha desenvolupat. El projecte es desenvolupa per una única persona, amb el suport i seguiment periòdic d'un tutor acadèmic, mitjançant reunions regulars destinades a la resolució de dubtes, la revisió de l'estat del projecte i la redefinició puntual d'objectius; aquest escenari s'ajusta especialment bé a metodologies àgils orientades a equips reduïts i amb un alt grau de flexibilitat.

Crystal [1] és una metodologia àgil, caracteritzada per la seva adaptabilitat al context del projecte i a les persones que hi participen, prioritzant la comunicació, la col·laboració i la simplicitat per sobre de processos rígids o excessivament formals. Dins la família de metodologies Crystal, tenim diverses categories segons la quantitat de persones que formen l'equip de treball, en aquest cas hem escollit Crystal Clear, ja que està pensada específicament per a projectes desenvolupats per equips molt petits, com un format per l'alumne i el tutor.

Els principis fonamentals de Crystal Clear que s'han aplicat en aquest projecte són:

- Enfocament centrat en les persones, adaptant la metodologia a les necessitats reals del desenvolupament.
- Realització d'entregues parcials i freqüents.
- Revisions periòdiques del progrés del projecte.
- Documentació lleugera però suficient per garantir la comprensió i seguiment del treball.

De manera complementària, he decidit utilitzar una estratègia de prototipat incremental [2], basada en la construcció progressiva del sistema mitjançant versions successives del prototip. Cada iteració ha donat lloc a un sistema funcional per si mateix, sobre el qual a cada fase posterior s'han anat afegint noves funcionalitats que feien el sistema més complet i es milloraven les existents a partir del feedback obtingut per part del tutor. Les fases del prototipat del meu sistema són les següents:

- Primera fase - Funcionament del sensor de gotes d'aigua i dels actuadors
- Segona fase - Integració del sistema mitjançant l'establiment de la comunicació entre plaques (versió mínima funcional del prototip)
- Tercera fase - Connexió a plataforma domètica del prototip
- Fase final - Millores i optimització

Aquesta aproximació ha reduït el risc de fracassar o tenir grans problemes en el desenvolupament del prototip a causa de fixar unes expectatives o objectius de disseny massa optimistes.

La combinació de Crystal Clear i prototipat incremental ha resultat adequada i efectiva, ja que ha permès un seguiment constant del projecte, la reducció de l'impacte dels obstacles i canvis produïts i també ha garantit l'obtenció progressiva d'un sistema encastrat funcional, escalable i alineat amb els requisits inicials del projecte.

4 MARC TEÒRIC

En aquesta secció es procedirà a explicar els conceptes clau per a entendre el desenvolupament del sistema.

4.1 Thread

Com a protocol de comunicació del prototip he fet servir Thread [3], que és un protocol de xarxa emergent orientat a aplicacions de domòtica i Internet de les Coses (IoT), dissenyat per proporcionar comunicacions fiables, segures i de baix consum energètic entre dispositius. El seu ús està en ascens gràcies a que utilitza IPv6, facilitant la integració amb xarxes IP i plataformes cloud, això esdevé un avantatge enfront altres protocols ja establerts com Zigbee, sumat al fet que manté els mateixos avantatges com la topologia de xarxa en malla amb capacitat d'autoreparació i un baix consum energètic com s'ha mencionat anteriorment.

Sota la capa de xarxa, el protocol Thread es construeix sobre l'estàndard IEEE 802.15.4, que defineix les capes física i MAC per a comunicacions sense fils de baixa potència i consum energètic, especialment adequades per a sistemes encastats.

La connexió entre una xarxa Thread i altres xarxes, com ara una xarxa domèstica basada en Wi-Fi o Ethernet, es realitza mitjançant un element anomenat Thread Border Router. Aquest component actua com a pont entre la xarxa Thread i la infraestructura IP, permetent la comunicació amb plataformes domòtiques i serveis externs. Un únic Border Router pot donar servei a múltiples nodes Thread, fet que permet construir sistemes escalables.

En molts sistemes, el Border Router es divideix conceptualment en dues parts: el Host, encarregat de gestionar la connectivitat IP i l'execució de la pila de xarxa, i el Radio Co-Processor (RCP) [4], responsable de la comunicació per ràdio mitjançant IEEE 802.15.4. La comunicació entre els dos es realitza mitjançant el protocol Spinel, que permet que el RCP transmeti les trames de ràdio cap al Host i que aquest les pugui processar per reconstruir els paquets de nivell superior. A la figura 1 es mostra el funcionament de Thread.

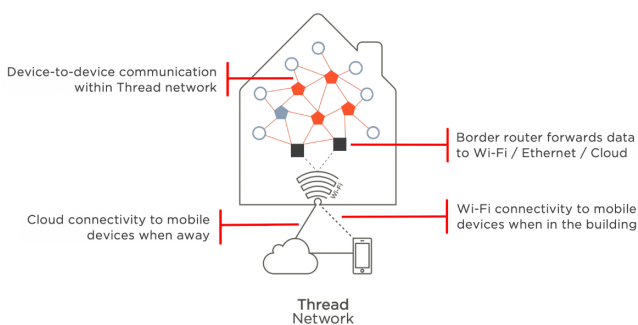


Fig. 1: Rol del Border Router

En conjunt, Thread i el Border Router ofereixen una solució estandaritzada, eficient i escalable per a la interconnexió de dispositius domòtics dins d'una llar intel·ligent.

4.2 Home Assistant

Home Assistant [5] és la plataforma de domòtica de codi obert més utilitzada a nivell mundial, actuant com a nucli centralitzat per a la integració, supervisió i automatització de dispositius d'una casa intel·ligent. Té un disseny orientat a la interoperabilitat, fet que permet gestionar dispositius

de diferents fabricants i tecnologies sota una mateixa infraestructura, sense dependència obligatòria de serveis cloud externs.

La plataforma s'executa de manera local, ja sigui de forma nativa o mitjançant virtualització, fet que permet que el processament i la gestió de l'estat dels dispositius es realitzi dins de la xarxa domèstica. Aquest enfocament millora la latència, la fiabilitat i la privacitat del sistema. També és important dir que Home Assistant suporta múltiples protocols habituals en entorns IoT, com ara MQTT, Wi-Fi, Ethernet, Zigbee o Thread, mitjançant un sistema d'integracions modulars.

S'ha escollit Home Assistant com a plataforma domòtica per la seva flexibilitat, ampli suport de protocols estàndard i gran adopció en entorns reals, fet que permet validar el prototip en un context proper a una instal·lació domòtica real.

4.3 MQTT

MQTT [6] (Message Queuing Telemetry Transport) és un protocol de missatgeria lleuger que es basa en el model publicador-subscriptor i està dissenyat específicament per a entorns amb dispositius amb recursos limitats i amplades de banda reduïdes, com és el cas de les cases domotitzades. En aquest model els dispositius no es comuniquen directament entre ells, sinó que publiquen i reben missatges a través d'un element central anomenat broker. Els dispositius publicadors envien missatges associats a un topic, mentre que els dispositius subscriptors reben únicament els missatges dels topics als quals estan subscriptes. Aquesta arquitectura fomenta l'escalabilitat i facilita la integració de nous dispositius sense modificar els ja configurats, a la figura 2 podem veure el funcionament.

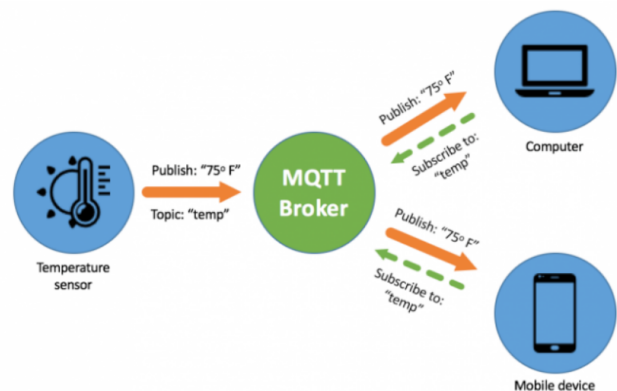


Fig. 2: Funcionament del Broker MQTT

MQTT ofereix diferents nivells de qualitat de servei (QoS), que permeten adaptar la fiabilitat de la comunicació segons els requisits de l'aplicació, des d'enviaments sense confirmació fins a lliuraments garantits. A més, el seu disseny minimalista redueix el consum energètic i la càrrega de processament, aspectes clau en sistemes encastats. En el context d'aquest projecte, s'ha escollit MQTT com a mecanisme de comunicació entre el Thread Border Router i la plataforma Home Assistant.

4.4 RTOS

Per acabar el marc teòric introduiré RTOS. Un Real-Time Operating System (RTOS) [7] és un sistema operatiu lleuger molt utilitzat en sistemes embastats, ja que aquests requereixen un comportament temporal previsible i RTOS ho proporciona. La seva funció principal és crear i gestionar múltiples tasques de manera concurrent mitjançant un planificador (scheduler), que assigna els recursos del sistema segons prioritats i restriccions temporals.

La característica més rellevant d'un RTOS és el determinisme, que garanteix temps de resposta acotats davant esdeveniments crítics, juntament amb una latència baixa, essencial en aplicacions de temps real. A més, permet una gestió eficient de prioritats, assegurant que les tasques més crítiques s'executin abans que les menys rellevants.

Els RTOS proporcionen mecanismes de concurrència i sincronització, com cues i locks, que permeten coordinar tasques i evitar condicions de carrera en sistemes multitasca. Finalment, cal destacar que estan optimitzats per funcionar amb recursos limitats, fet que els fa especialment adequats per a dispositius IoT.

5 MATERIAL

En aquesta secció es presenta el material emprat en el desenvolupament del prototip.

5.1 Hardware

- 1 placa ESP32-S3-DevKitC-1 [8]. Connectivitat Wi-Fi integrada, fa de Host del Thread Border Router i és on es connecten els actuadors (pantalla i buzzer).
- 2 plaques ESP32-H2-DevKitM-1 [9]. Suporten IEEE 802.15.4, permetent la comunicació per Thread, una fa de RCP del Border Router i l'altra estarà col·locada en el lloc d'interès amb el sensor d'aigua.
- mòdul sensor de gotes d'aigua YL-83 [10].
- Pantalla ST7789 SPI TFT 2.0" 320 x 240 [11].
- Buzzer passiu PS1240P02BT [12].
- 11 cables DuPont femella-femella.
- 8 cables DuPont mascle-femella.
- Bateria LiPO 3.7 V 1500mAh.

5.2 Software

- OpenThread. Implementació de codi obert de Thread.
- FreeRTOS. Implementació de codi obert capdavantera d'un RTOS.
- ESP-IDF. SDK oficial per plaques ESP.
- Mosquitto [13]. Popular broker MQTT molt eficient.
- Virtualbox. Màquina virtual Linux on s'executa Home Assistant.
- Home Assistant.

- VS Code.
- Llenguatge C.

6 DESENVOLUPAMENT

El desenvolupament del prototip s'ha dut a terme seguint l'enfocament de prototipat incremental abans comentat. Aquest enfocament ha permès dividir el projecte en diverses fases successives, les quals, menys la primera, cadascuna donava lloc a una versió funcional del sistema que amplia i millorava l'anterior, resultant en un sistema cada vegada més versàtil.

6.1 Primera fase - Implementació dels elements bàsics del sistema

La primera fase del desenvolupament s'ha centrat en la implementació i validació dels elements hardware principals del sistema de manera independent, sense encara introduir comunicacions de xarxa. En aquesta fase s'ha implementat sobre una placa ESP32-S3 el control dels actuadors locals, concretament un buzzer i una pantalla LCD.

Per la pantalla s'ha realitzat una tasca en la que s'apaga i s'encén la il·luminació durant 30 segons, mostrant intermitentment una imatge d'una gota d'aigua, la qual s'ha descodificat a mitja resolució per reduir l'ocupació de memòria i reescalat fent ús de Nearest Neighbor [14], la funció d'aquest algoritme s'explica a la figura 3.

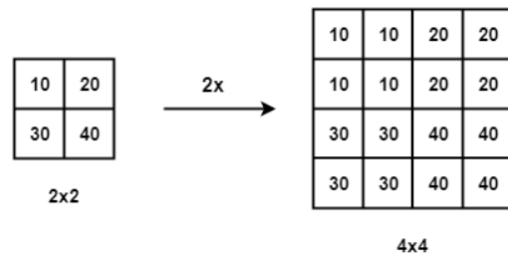


Fig. 3: Algoritme Nearest Neighbor per reescalat d'imatges

Per al buzzer s'han generat polsos gràcies a RMT (Remote Control Transceiver) utilitzant una codificació musical de parells freqüència-durada, creant una seqüència de sons aguts llargs i sons greus curts que formen una alarma. S'ha obtingut un correcte funcionament simultani de l'alarma i la pantalla a la ESP32-S3 degut al funcionament multitasca RTOS.

En una de les plaques ESP32-H2, s'ha desenvolupat la connexió mitjançant GPIO al sensor d'aigua, realitzant lectures periòdiques de l'estat del sensor. La connexió passa per una placa de control amb el xip comparador LM393 que incorpora el sensor.

L'objectiu d'aquesta primera iteració ha estat verificar el correcte funcionament individual de cada component, així com establir una base funcional sòlida sobre la qual construir les fases posteriors del sistema.

6.2 Segona fase - Border Router i comunicació Thread

Un cop validada la detecció d'aigua i el funcionament dels actuadors locals, la segona fase del projecte s'ha centrat en dotar el sistema de comunicació Thread i en la implementació d'un Border Router funcional. Per a aquesta fase s'ha desenvolupat un Border Router basat en dues plaques ESP32, separant clarament les funcions de comunicació ràdio i de gestió de xarxa.

La placa ESP32-H2 que encara no havia estat utilitzada s'ha configurat com a Radio Co-Processor (RCP), encarregant-se exclusivament de la recepció i transmissió de les trames IEEE 802.15.4. La placa ESP32-S3 actua com a host del Border Router, executant la pila OpenThread [15], gestionant la xarxa Thread i establint la connexió amb la xarxa IP mitjançant Wi-Fi.

La comunicació entre ambdues plaques s'ha implementat mitjançant el protocol Spinel, que permet transferir les dades de ràdio del RCP cap al host a través d'una connexió sèrie UART. Aquesta arquitectura fa possible disposar d'un Border Router complet tot i que la placa principal no disposi de transceptor IEEE 802.15.4.

A nivell de codi, el host del Border Router inicialitza la pila OpenThread i crea una interfície de xarxa virtual que representa la xarxa Thread dins del sistema, permetent l'assignació automàtica d'adreces IPv6. Aquesta interfície es vincula amb la connexió Wi-Fi del dispositiu, que actua com a xarxa troncal (backbone), permetent així l'encaminament transparent de paquets entre la xarxa Thread i la xarxa IP.

Un cop creat el Border Router, s'ha establert la comunicació entre la placa amb el sensor d'aigua i aquest mitjançant missatges UDP sobre Thread. Quan hi ha una detecció d'aigua, la placa amb el sensor envia un missatge UDP cap al Border Router, que el rep, el processa i activa els actuadors locals ja mencionats. Per a la recepció d'aquests missatges s'ha implementat un socket UDP [16] utilitzant l'API d'OpenThread, vinculat a un port específic.

Per garantir un funcionament robust i evitar bloquejos, la recepció dels missatges UDP s'ha gestionat mitjançant cues de FreeRTOS [17]. El handler de recepció s'encarrega únicament de copiar el missatge rebut a una cua, mentre que una tasca independent processa posteriorment l'alarma i executa les accions associades. Aquest enfocament permet desacoblar la comunicació de xarxa del processament de l'alarma i assegura una execució eficient del sistema.

Com a resultat d'aquesta segona fase s'obté una primera versió funcional mínima del prototip, capaç de detectar una fuga d'aigua, transmetre l'alarma a través de la xarxa Thread i generar una resposta local. Aquesta iteració valida tant la comunicació entre dispositius com l'arquitectura distribuïda escollida, establint una base sòlida per a les fases posteriors del projecte.

6.3 Tercera fase - Integració amb Home Assistant mitjançant MQTT

Un cop validada la comunicació local mitjançant Thread, la tercera fase del projecte s'ha centrat en ampliar el sistema

amb capacitats d'alerta remota, mitjançant la seva integració amb una plataforma domòtica.

Per assolir aquest objectiu, s'ha incorporat el protocol de missatgeria MQTT al Border Router, permetent la publicació d'alertes cap a un broker MQTT ubicat a la xarxa local. El Border Router actua com a client MQTT, establint una connexió persistent amb el broker. Quan es detecta una fuga d'aigua, el sistema publica un missatge d'alarma en un tòpic específic, i posteriorment restableix l'estat quan l'alarma finalitza.

El broker MQTT s'ha integrat amb Home Assistant [18], que actua com a plataforma centralitzada de supervisió. Dins d'aquesta plataforma s'ha definit un sensor binari associat a la detecció de fuites d'aigua, el qual reflecteix l'estat del sistema en temps real.

Adicionalment, s'ha configurat un mecanisme de disponibilitat que permet a Home Assistant conèixer si el dispositiu es troba connectat o fora de servei, millorant la robustesa del sistema. A partir d'aquest sensor, s'ha implementat també una automatització que permet l'enviament de notificacions immediates als dispositius de l'usuari que tinguin l'aplicació Home Assistant, com ara telèfons mòbils o ordinadors.

A continuació podem veure a la figura 4 la col·locació al model de capes OSI [19] dels protocols implementats al Thread Border Router.

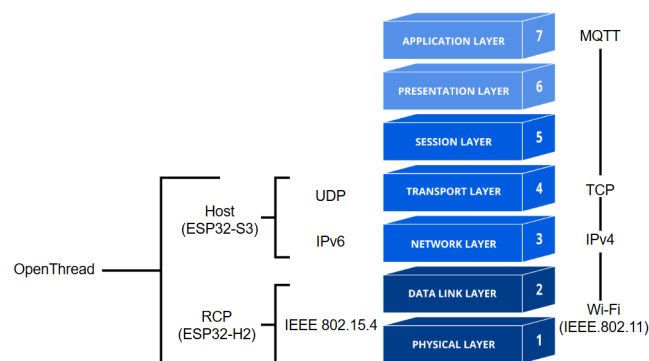


Fig. 4: Protocols del Border Router a les capes OSI

Gràcies a aquesta integració, el prototip evoluciona d'un sistema amb alerta local a un sistema domòtic complet, capaç de proporcionar monitoratge i avisos remots en temps real. Aquesta fase demostra la capacitat del Border Router per actuar com a passarel·la entre la xarxa Thread i serveis IP estàndard, incrementant significativament la utilitat pràctica del sistema en entorns reals.

6.4 Fase final - Millores i optimització del sistema

Finalment, la quarta fase del projecte s'ha dedicat a la millora de la robustesa i fiabilitat del sistema, així com a l'optimització de l'ús de recursos, amb l'objectiu d'obtenir una versió final més estable i adequada per a un ús realista.

Una de les millores principals ha estat la incorporació d'un filtre software en la detecció d'aigua. Aquest mecanisme es basa en la validació de múltiples lectures positives consecutives abans de generar una alerta, reduint així

la probabilitat de falses deteccions degudes a soroll, humitat ambiental puntual o interferències elèctriques. Aquesta estratègia incrementa la fiabilitat del sistema sense afegir complexitat al maquinari.

Paral·lelament, s'han dut a terme diverses tasques d'optimització a la placa ESP32-S3, que és la que té la major càrrega de processament i la major ocupació de memòria. Entre aquestes optimitzacions destaquen la reducció de la memòria assignada a cada tasca, l'ajust dels paràmetres de compilació per afavorir la generació de codi més compacte i l'eliminació de logs generats; que són útils en fases de desenvolupament i depuració però innecessaris en el context d'un prototip orientat a l'usuari final.

Com a resultat d'aquesta última fase, s'obté la versió final del prototip, caracteritzada per un funcionament més robust, un ús més eficient dels recursos i una major preparació per a una possible ampliació o desplegament en entorns domèstics reals.

7 RESULTATS

7.1 Arquitectura final

En la figura 5 es presenta l'arquitectura aconseguida.

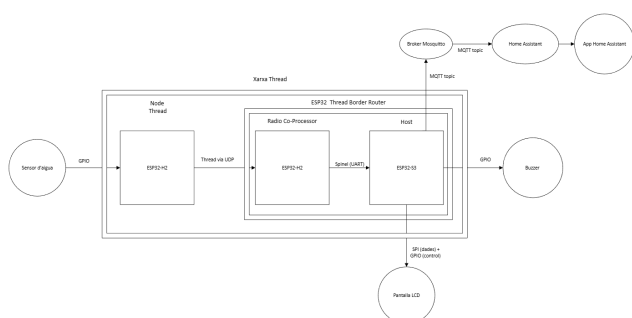


Fig. 5: Arquitectura global del sistema

7.2 Punts destacats

El sistema implementat presenta diversos punts forts destacables:

- Una arquitectura distribuïda i escalable, basada en el protocol Thread, que permet afegir nous sensors a la xarxa de manera senzilla.
- Sistema d'alerta dual, tenim notificació local mitjançant actuadors (pantalla i buzzer) i notificació remota mitjançant Home Assistant.
- Ús de tecnologies estàndard i obertes, com Thread, UDP, MQTT i RTOS, que garanteixen interoperabilitat i faciliten futures ampliacions.
- Comunicació robusta gràcies a un desacoblament entre la recepció de missatges i el seu processament mitjançant mecanismes de concurrència de FreeRTOS.
- Baix consum energètic en la placa amb el sensor gràcies a l'ús de IEEE 802.15.4 i UDP.

7.3 Evidències

A continuació es presenten les evidències del treball realitzat, seguint l'ordre del flux de funcionament del sistema quan es detecta l'aigua.

Detecció d'aigua i enviament de la alarma:

```
I(581) OPENTHREAD:[N] Mle-----: Role detached -> router
I(581) OPENTHREAD:[N] Mle-----: Partition ID 0x28fb88d9

(691) OT_STATE: Set dns server address: FDBD:2420:9C:2::808:808
> W (64771) SENSOR: ¡FUGA CONFIRMADA! (5/5 lecturas) Enviando alarma...
I (64771) UDP_SEND: Alarma enviada a fd00:db8:a0:0:b422:1bc0:5b22:50ea
I (91771) SENSOR: Sensor seco de nuevo
```

Fig. 6: Logs de detecció d'aigua i enviament de la alarma

Es pot comprovar a la figura 6 que el rol de la placa és "router", ja que un node Thread pot reencaminar paquets a un altre node, formant una xarxa en malla autorreparable, tal com havíem vist.

Funcionament del Border Router:

```
I (10040) OT_STATE: netif up
I (10050) OPENTHREAD: NAT64 ready
I (10050) OPENTHREAD: Platform UDP bound to port 61631
I (20650) esp_ot_br: MQTT conectado al broker
I (36630) OPENTHREAD:[N] Mle-----: RLOC16 8400 -> fffe
I (36670) OPENTHREAD:[N] Mle-----: Attach attempt 1, AnyPartition reattaching with Active Dataset
I (43280) OPENTHREAD:[N] RouterTable---: Allocate router id 33
I (43280) OPENTHREAD:[N] Mle-----: RLOC16 fffe -> 8400
I (43290) OPENTHREAD:[N] Mle-----: Role detached -> leader
I (43290) OPENTHREAD:[N] Mle-----: Partition ID 0x568cf85
I (43310) OPENTHREAD: Platform UDP bound to port 49154
I (54470) example_connect: got IPv6 event: interface "example_netif_sta" address: fdde:adb0:beef:cafe::2cf:13ff:fe1a:22fc, type: ESP_IP6_ADDR_IS_UNIQUE_LOCAL
I (93460) OPENTHREAD:[N] RouterTable---: Allocate router id 52
I (126750) UDP_PROC: Mensaje desde [fd00:0db8:00a0:0000:abb1:775e:07ae:9913]:49155 -> FUGA_DETECTADA
I (126750) UDP_PROC: Inicializando LCD...
I (126940) esp_ot_br: LCD inicializado correctamente (backlight apagado)
I (126950) esp_ot_br: Alerta MQTT publicada (msg_id=3441)
I (136010) esp_ot_br: Secuencia del buzzer completada
I (157010) esp_ot_br: Alerta MQTT reseteada a OFF (msg_id=24996)
I (157010) UDP_PROC: Backlight apagado - pantalla lista para próxima alarma
```

Fig. 7: Logs de funcionament del Border Router

En relació al que s'ha dit a la figura 6, a la figura 7 podem observar que el Border Router té el rol de Headercom a creador i administrador de la xarxa Thread.

A continuació, a les dues següents figures es presenten els dos estats del sensor en Home Assistant

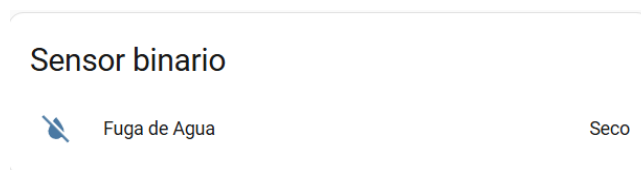


Fig. 8: Estat sec a Home Assistant

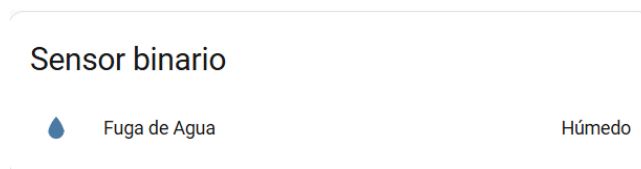


Fig. 9: Estat humit a Home Assistant

Finalment, es procedeix a mostrar a la figura 10 la notificació en el telèfon mòbil, mostrant que podem obtenir l'alerta de manera remota.

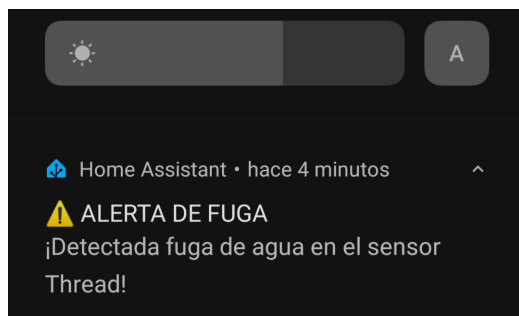


Fig. 10: Notificació remota a la app

7.4 Proves de comunicació

Adicionalment, s'han realitzat proves a diferents distàncies per validar la comunicació entre el node sensor i el Border Router, amb l'objectiu de verificar tant l'estabilitat de la transmissió com l'abast del sistema en un entorn domèstic real.

La distància màxima provada ha estat d'aproximadament 30 metres, sense línia de visió directa, ja que les proves s'han dut a terme dins d'un habitatge amb diferents estances i obstacles físics. En aquestes condicions, el sistema ha mantingut una comunicació estable, amb recepció correcta dels missatges d'alarma enviats pel node sensor cap al Border Router. Amb això confirmem també la capacitat d'ampliació del sistema amb més nodes amb sensors repartits per una casa.

A més de la recepció del missatge, s'ha confirmat que l'alarma local, generada mitjançant el buzzer connectat a la placa del Border Router, és perfectament audible des d'aquesta distància, reforçant la utilitat del sistema com a mecanisme d'avís immediat dins de l'habitatge.

Tot això valida la idoneïtat del protocol Thread per a comunicacions domòtiques de curt i mitjà abast, i la funció del buzzer com alerta local en condicions realistes d'ús.

8 LÍNIA DE MILLORA - MATTER

Una possible línia de millora del sistema desenvolupat consisteix en la integració del protocol Matter [20] sobre la infraestructura Thread ja implementada. Matter és un estàndard obert impulsat per la Connectivity Standards Alliance (CSA), dissenyat per garantir la interoperabilitat entre dispositius domòtics de diferents fabricants i plataformes, com ara Home Assistant, Google Home, Apple HomeKit o Amazon Alexa.

Matter funciona a nivell d'aplicació i se situa per sobre de protocols de xarxa com Thread. En el cas del meu projecte, Thread ja proporciona una base sòlida de comunicació IP segura i de baix consum, fet que converteix el prototip en un candidat idoni per a l'execució de Matter sobre Thread.

La incorporació de Matter al Border Router permetria que el sistema fos reconegut com un dispositiu domòtic

estàndard dins d'ecosistemes comercials, facilitant la seva integració directa en llars domotitzades sense necessitat de configuracions específiques ni integracions personalitzades com MQTT. Això inclouria processos de comissionament estandaritzats mitjançant codis QR, descoberta automàtica de dispositius i una gestió unificada de seguretat i identitat.

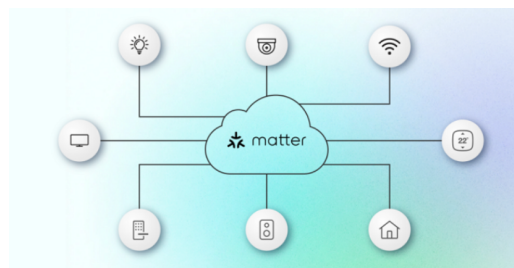


Fig. 11: Concepte de Matter

Tot i els avantatges esmentats, la implementació de Matter implicava un afegit al sistema amb una complexitat tècnica fora de l'abast del TFG, sobretot tenint en compte el temps disponible; llavors s'ha decidit implementar MQTT que és més lleuger i ja em permet connectar el sistema a Home Assistant.

El sistema desenvolupat estableix una base tècnica robusta que permetria, en una futura evolució, incorporar Matter al Border Router i transformar el prototip en un dispositiu plenament compatible amb entorns domòtics comercials, dotant el dispositiu d'una interoperabilitat total.

9 CONCLUSIONS

En conclusió, he implementat amb èxit un prototip domòtic de detecció de fuites d'aigua basat en una arquitectura distribuïda i en l'ús de protocols estàndard de l'àmbit IoT. El sistema desenvolupat és capaç de detectar una fuga d'aigua, transmetre l'alerta mitjançant una xarxa Thread i generar avisos tant locals com remots, complint així els objectius principals plantejats a l'inici del projecte.

La prova de distància ha evidenciat que el prototip està preparat per al seu desplegament en cases, com a mínim petites i mitjanes, i és fàcilment ampliable per a tenir més nodes sensors (fins i tot amb altres tipus de sensors).

La metodologia escollida ha resultat especialment adequada per al desenvolupament del projecte, ja que m'ha permès obtenir versions funcionals parcials des de fases inicials, reduint el risc de no assolir un prototip operatiu al final del treball. Les revisions periòdiques amb el tutor i la flexibilitat de la metodologia han estat clau per reajustar objectius i prendre decisions tècniques al llarg del desenvolupament.

Des del punt de vista tecnològic, el projecte ha permès treballar de manera pràctica amb protocols i estàndards oberts com IEEE 802.15.4, Ipv6, Thread, UDP i MQTT. L'elecció de UDP sobre TCP a la xarxa Thread ve donada perquè ofereix menor latència i consum energètic, aspectes que considero importants en el sistema. A més, l'ús de Thread complementa les limitacions inherents a UDP mitjançant mecanismes de seguretat a nivell MAC i una topologia en malla que aporta robustesa a la comunicació.

Durant el desenvolupament han sorgit diverses dificultats tècniques que han contribuït significativament a l'aprenen-

tatge. la més rellevant ha estat la impossibilitat d'integrar directament un sensor comercial desenvolupat amb Thread a la xarxa creada, fet que ha posat de manifest que molts dispositius comercials utilitzen Matter com a capa d'aplicació sobre Thread. Aquesta limitació ha permès entendre amb més profunditat la separació de responsabilitats entre Thread i Matter dins del model de capes OSI, així com la importància de Matter en la interoperabilitat de sistemes domòtics comercials. També cal destacar la gestió de la concurrència al Border Router, on inicialment un ús incorrecte de mecanismes de sincronització i una càrrega excessiva en determinades tasques van provocar bloquejos del sistema. La resolució d'aquest problema ha reforçat la importància de dissenyar correctament l'arquitectura de tasques en un RTOS, separant clarament la comunicació de xarxa de l'activació de les alertes i utilitzant adequadament cues i mecanismes de protecció per garantir l'estabilitat del sistema.

La integració final amb Home Assistant mitjançant MQTT ha ampliat considerablement les capacitats del prototip, permetent la visualització remota de l'estat del sensor i l'enviament de notificacions en temps real. Aquesta integració demostra que el sistema no només és funcional a nivell local, sinó que pot formar part d'un ecosistema domòtic més ampli sense necessitat de modificar la infraestructura Thread existent.

Finalment, cal destacar que el prototip s'ha desenvolupat utilitzant components de baix cost, com plaques amb memòria ajustada i actuadors senzills. Això demostra que és possible assolir solucions eficients i robustes sense recórrer a elements complexos o costosos, fet que maximitza els beneficis en el context d'un producte real.

En conjunt, els resultats obtinguts validen l'enfocament escollit i estableixen una base sòlida per a futures ampliacions i millores del sistema, i el desenvolupament del projecte m'ha permès agafar pràctica en el disseny de sistemes encastats, integrant coneixements de xarxes, protocols de comunicació tant sèrie com sense fils, RTOS i plataformes domòtiques, i també millorar la resolució de problemes i treballar la creativitat per a crear solucions efectives.

10 ÚS DE IA GENERATIVA

En el transcurs del desenvolupament del projecte he utilitzat la IA per als següents casos:

- Generació d'una imatge al meu gust d'una gota d'aigua en una pàgina generadora d'imatges sense Copyright.
- Ajuda per a entendre més ràpid perquè en un moment concret del desenvolupament se'm bloquejava el Border Router i alguns problemes de compilació.
- Millores en la redacció d'aquest informe i el format de les referències.

11 AGRAÏMENTS

Agrair al meu tutor Màrius Montón Macian, primer per la disponibilitat a fer reunions setmanals per a la tutorització

del projecte, i segon, pels seus consells i constants idees de millora del sistema, el qual inicialment anava a ser molt més bàsic.

A la Universitat Autònoma de Barcelona i els seus professors per tots els coneixements impartits aquests anys.

I finalment, a la meua mare i la meua parella, que sempre han estat al meu costat per donar-me suport i treure'm un somriure, i al meu pare i el meu germà, que encara que ja no estiguin, els portaré en el cor sempre i gràcies a ells soc enginyer informàtic, el meu pare per ensenyar-me el que és un ordinador de petit, i el meu germà per pegar-me la seva creativitat i ganes de crear coses.

REFERÈNCIES

- [1] IEBS Digital School, "Qué es Crystal Agile Framework: una metodología que se adapta a tu equipo," IEBS Blog, 2022. [Online]. Available: <https://www.iebschool.com/hub/que-es-crystal-agile-framework-una-metodologia-que-se-adapta-a-tu-equipo-agile-scrum/>
- [2] Proyectos Ágiles, "Desarrollo iterativo-incremental," ProyectosAgiles.org, 2020. [Online]. Available: <https://proyectosagiles.org/desarrollo-iterativo-incremental/>
- [3] Aqara, "¿Qué es el protocolo Thread y cómo funciona?," Aqara EU Blog, 2023. [Online]. Available: <https://eu.aqara.com/es-eu/blogs/noticias/que-es-el-protocolo-thread-y-como-functiona>
- [4] OpenThread, "Co-Processor Platforms," OpenThread Documentation. [Online]. Available: <https://openthread.io/platforms/co-processor>
- [5] Home Assistant, "Home Assistant Documentation," home-assistant.io, 2024. [Online]. Available: <https://www.home-assistant.io/docs/>
- [6] MQTT, "MQTT: The Standard for IoT Messaging," mqtt.org. [Online]. Available: <https://mqtt.org/>
- [7] Universitat Carlemany, "Sistema operativo en tiempo real (RTOS): qué es y para qué sirve," Universitat Carlemany Blog, 2021. [Online]. Available: <https://www.universitatcarlemany.com/actualidad/blog/sistema-operativo-en-tiempo-real/>
- [8] Espressif Systems, ESP32-S3 DevKitC-1 User Guide, 2024. [Online]. Available: <https://docs.espressif.com/projects/esp-dev-kits/en/latest/esp32s3/esp32-s3-devkitc-1/index.html>
- [9] Espressif Systems, ESP32-H2 DevKitM-1 User Guide, 2024. [Online]. Available: <https://docs.espressif.com/projects/esp-dev-kits/en/latest/esp32h2/esp32-h2-devkitm-1/>
- [10] Maker Advisor, "YL-83 Rain Sensor: Tools, Features and How It Works," MakerAdvisor.com, 2023. [Online]. Available: <https://makeradvisor.com/tools/yl-83-rain-sensor/>

- [11] Sitronix, ST7789 LCD Controller Datasheet, 2021. [Online]. Available: <https://www.alldatasheet.es/datasheet-pdf/download/2200467/SITRONIX/ST7789.html>
- [12] TDK Corporation, “PS1240P02BT Piezoelectric Buzzer,” Digi-Key, 2024. [Online]. Available: <https://www.digikey.es/en/products/detail/tdk-corporation/PS1240P02BT/935924>
- [13] Eclipse Foundation, “Eclipse Mosquitto MQTT Broker,” mosquitto.org. [Online]. Available: <https://mosquitto.org/>
- [14] ScienceDirect, “Nearest Neighbor Interpolation,” ScienceDirect Topics. [Online]. Available: <https://www.sciencedirect.com/topics/engineering/nearest-neighbor-interpolation>
- [15] M. Mihaeljans and A. Skrastins, “OpenThread Network Density Evaluation: Quantitative Analysis,” Proc. 2023 Symposium on Internet of Things (SIoT), São Paulo, Brazil, 2023.
- [16] OpenThread, “UDP API Reference,” OpenThread Documentation. [Online]. Available: <https://openthread.io/reference/group/api-udp>
- [17] FreeRTOS, “Queues, mutexes and semaphores,” FreeRTOS Kernel Documentation. [Online]. Available: <https://freertos.org/Documentation/02-Kernel/02-Kernel-features/02-Queues-mutexes-and-semaphores/01-Queues>
- [18] Home Assistant, “MQTT Integration,” home-assistant.io, 2024. [Online]. Available: <https://www.home-assistant.io/integrations/mqtt/>
- [19] IBM, “OSI Model,” IBM Think Topics. [Online]. Available: <https://www.ibm.com/es-es/think/topics/osi-model>
- [20] TP-Link, “What is Matter and How Does It Work?,” TP-Link Technology, 2023. [Online]. Available: <https://www.tp-link.com/es/technology/matter/>

APÈNDIX

A.1 Imatges del prototip

A continuació es presenten les imatges de les dues parts del prototip, el Border Router format per la unió de la placa ESP32-H2 i ESP32-S3 (amb els actuadors), i l'altra placa ESP32-H2 (amb el sensor).

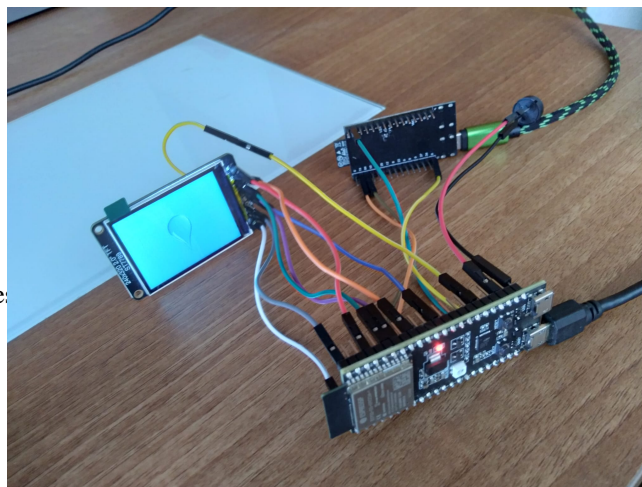


Fig. 12: Imatge del Border Router i actuadors

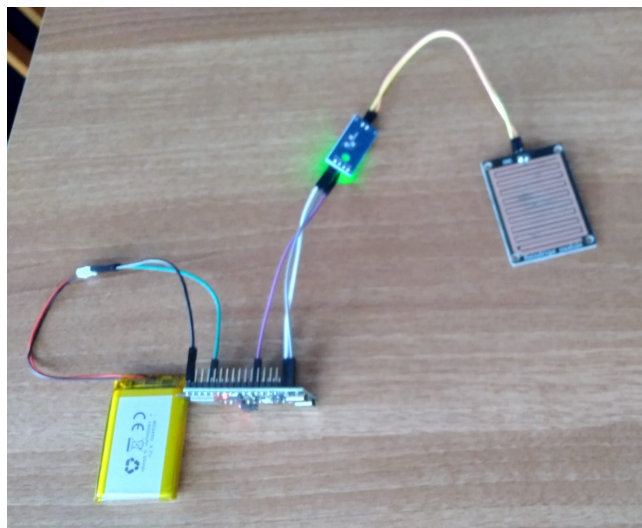


Fig. 13: Imatge de la ESP32-H2 amb el sensor