

Sistema de control i monitorització de curses Slot basat en una Raspberry Pi, Contiki i Bluetooth Low Energy

Albert Babí Oller - Universitat Oberta de Catalunya (UOC)
F. Xavier Moncunill Geniz - Universitat Politècnica de Catalunya (UPC)
Email: ababi92@uoc.edu

Resum—Aquest article presenta una plataforma unificada que ofereix les funcionalitats bàsiques per a monitoritzar i controlar curses Slot. El disseny és una proposta innovadora que utilitza tecnologies relacionades amb l'àmbit emergent d'Internet de les Coses. El sistema ofereix un control remot dels cotxes Slot basat en el protocol de comunicació eficient Bluetooth Low Energy. També permet monitoritzar les curses des d'una interfície gràfica gràcies a nodes de detecció de moviment situats al llarg de la pista, que utilitzen el sistema operatiu Contiki per a enviar informació a través del protocol 802.15.4. Els resultats de les curses es poden emmagatzemar en una base de dades MySQL i consultar posteriorment des de la interfície gràfica o una pàgina web. Per tal d'avaluar el disseny s'ha implementat un prototip en una Raspberry Pi amb Debian.

Paraules Clau—Raspberry Pi, Contiki, Bluetooth Low Energy, curses Slot, Internet of Things

I. INTRODUCCIÓ

La revolució d'Internet de les Coses, de l'anglès Internet of Things (IoT), ha representat un factor clau en el desenvolupament tecnològic global dels últims anys. L'intercanvi d'informació entre dispositius de recursos reduïts ha fet possible una gran diversitat d'aplicacions en àmbits com la domòtica, la medicina o bé l'electrònica de consum, entre d'altres [1].

Tanmateix, a mesura que l'univers d'IoT creix, apareixen també més reptes per a resoldre. El mercat es troba fragmentat a causa de l'alta competitivitat entre les grans corporacions, el hardware desenvolupat és heterogeni i sovint tancat o propietari [2]. La utilització d'estàndards i tecnologies obertes és necessària per a fer possible la unificació de l'ecosistema d'IoT i facilitar l'escalabilitat i interoperabilitat entre la immensitat de solucions [3] [4].

Aquest article analitza un sector de l'electrònica de consum que ha sigut liderat per companyies comercials des dels seus inicis: les curses Slot. Des de la seva aparició en el mercat a inicis del segle XX [5] [6], la indústria ha progressat de forma constant, adoptant les diferents tecnologies de cada moment. Tot i així, les millores més rellevants han estat influenciades per l'evolució de les xarxes de comunicació sense fils. El control remot amb múltiples canals [7] [8] o streaming de vídeo [9] en temps real són exemples en els quals la tecnologia Slot i la comunicació sense fils han estat íntimament relacionades. Però a mesura que les companyies

comercials desenvolupen productes propietaris, els projectes de programari lliure esdevenen essencials per garantir que l'evolució de la tecnologia Slot sigui incremental i escalable.

Aquest article proposa un disseny unificat per a controlar i monitoritzar curses Slot basat exclusivament en hardware obert i protocols de comunicació sense fils estàndards. D'una banda, ofereix un control remot dels cotxes basat en la comunicació Bluetooth Low Energy (BLE). El control de la velocitat es pot implementar de diverses formes gràcies a l'abstracció d'una *Application Programming Interface* (API) flexible i oberta. D'altra banda, ofereix una interfície gràfica, en anglès Graphical User Interface (GUI), per a la monitorització en temps real. Aquesta consisteix en un semàfor, un comptador de voltes, un indicador de posició i un cronòmetre amb múltiples temps parcials per volta. Els parcials es calculen mitjançant nodes de detecció de moviment distribuïts al llarg de la pista Slot, els quals envien missatges al node principal a través del protocol de comunicacions de baixa potència 802.15.4 i el sistema operatiu Contiki. Opcionalment, es poden emmagatzemar els resultats de cada cursa per a ser consultats posteriorment des de la GUI o bé des d'internet a través d'una pàgina web.

El disseny ha sigut implementat i testejat en un prototip. El resultat és una alternativa als productes comercials que unifica les seves funcionalitats en una plataforma centralitzada. El fet que el sistema sigui lliure i obert facilita la posterior col·laboració de tercers per a un desenvolupament en comunitat, millorant la fiabilitat, funcionament i característiques del sistema.

La resta del document segueix la següent estructura: la Secció II presenta els protocols de comunicació sense fils de baix consum i analitza els requeriments del sistema. La Secció III en detalla la implementació. La Secció IV avalua el prototip implementat i inclou resultats experimentals. Finalment, la Secció V engloba les conclusions de l'article i exposa les línies de treball futures.

II. DISSENY DEL SISTEMA

Aquesta secció descriu el sistema des d'un punt de vista conceptual. Primer s'analitzen els protocols de comunicació sense fils òptims per a la proposta del disseny. Seguidament l'anàlisi de requeriments i casos d'ús identifica les funcionalitats, comportament i factors de qualitat del disseny. Finalment,

es descriu la seva arquitectura i components.

A. Protocols de comunicació sense fils de baixa potència

Les especificacions estàndard i els protocols de comunicació de baixa potència són essencials en el desenvolupament d'IoT, per exemple, en xarxes de sensors o sistemes encastats. Els sistemes han de ser eficients i flexibles per a facilitar la interconnexió entre tot tipus de dispositius (sovint de recursos reduïts), no només en xarxes locals sinó també a través de l'internet global [10].

A continuació s'analitzen tres dels protocols de comunicació eficient més populars [11] [12]. També es descriuen algunes de les tècniques utilitzades en IoT. Els protocols descrits ofereixen funcionalitats d'alt nivell per sobre d'estàndards IEEE (capa física i d'enllaç segons el model OSI).

1) *Bluetooth Classic vs Bluetooth Smart*: El Bluetooth Classic es basa en l'estàndard IEEE 802.15.1. Està dissenyat per a aplicacions de transmissió de curt abast, conegudes en anglès com Wireless Personal Area Networks (WPAN). El Bluetooth és adient per a transmissions contínues com ara l'*streaming* de música o l'enviament de fitxers degut a la seva alta velocitat de comunicació.

Durant el desenvolupament de la tecnologia Bluetooth s'han establert diverses versions (v1/2/3). Tanmateix, la última (v4) inclou millores i canvis significants i és considerada un protocol independent anomenat Bluetooth Smart o BLE [13]. Aquest suposa una millora substancial en el consum d'energia amb l'objectiu de satisfer les necessitats d'IoT. Per contra, la velocitat màxima de transmissió és més baixa (Taula I).

Ambdues versions de Bluetooth (Classic i Smart) es basen en tipologies de comunicació punt a punt.

2) *ZigBee i 6LoWPAN*: Basat en l'estàndard IEEE 802.15.4, ZigBee defineix una especificació per a xarxes WPAN de baixa velocitat, en anglès, *low-rate WPAN* (LR-WPAN). El protocol és adient en aplicacions de control i monitorització degut al baix consum i alta versatilitat [14].

Tal i com es pot observar en la Taula I, la velocitat de transmissió de ZigBee és menor que la de Bluetooth, però l'abast és major. Això ofereix més flexibilitat a l'hora de dissenyar xarxes més extenses. També cal tenir en compte que la tipologia que ofereix la xarxa ZigBee pot ser punt a punt, estrella o bé *mesh*.

6LoWPAN és, en anglès, l'acrònim de *IPv6 over Low Power WPAN* i, com el seu nom indica, l'objectiu és integrar qualsevol tipus de dispositiu a IoT a través de la transmissió de paquets IPv6 encapsulats en xarxes 802.15.4 [15].

El protocol 6LoWPAN pot ser implementat tant a través d'una xarxa ZigBee o bé utilitzant alternatives de software més complexes, com ara sistemes operatius per a dispositius de recursos reduïts.

3) *Contiki: el sistema operatiu obert per a IoT*: Durant els últims anys el desenvolupament de sistemes operatius ha mantingut un paper molt important en l'àmbit d'IoT [16]. La optimització de codi ha permès la implementació de software complex en sistemes encastats de recursos reduïts. Tot i que els sistemes operatius ofereixen un major grau de funcions que

	Bluetooth	BLE	ZigBee
Velocitat màx. (Mbps)	4	1	0.2 a 0.25
Potència màx. (dBm)	0 a 20	-20 a 10	-32 a 0
Dist. màx. (m)	10	10	100

TAULA I
COMPARATIVA ENTRE BLUETOOTH, BLE I ZIGBEE

els protocols i no s'inclouen en la mateixa categoria, alguns d'ells estan especialitzats en les xarxes de comunicacions, com a complement de protocols IEEE estàndards.

Contiki persegueix l'objectiu de convertir-se en el sistema operatiu *de-facto* per a IoT [17]. Algunes de les seves funcionalitats són el *multi-threading*, la gestió de memòria, una *Shell* interactiva o bé un stack de comunicació TCP/IP, entre d'altres. Aquest s'anomena uIP i facilita funcionalitats de 6LoWPAN com ara adreçament o encaminament TCP/IPv6.

L'impacte memòria del codi bàsic de Contiki és de 2kB per la RAM i 40 kB per la ROM [18].

B. Requirements no-funcionals

Les plataformes d'IoT comparteixen alguns requeriments no-funcionals que especifiquen les qualitats que el sistema ha d'assolir [2]. Aquesta disseny es centra en els següents:

1) *Escalabilitat*: El sistema ha de garantir l'intercanvi d'informació des de múltiples nodes. El funcionament en temps real ha de ser precís tot i les connexions i flux de dades variable.

Les millores i extensions del sistema han de ser possibles amb la mínima modificació i garantint en tot moment el correcte funcionament del sistema.

Un exemple d'ampliació seria afegir temps parcials per volta addicionals en el cronòmetre. Caldria afegir nodes de detecció de moviment sense representar un impacte considerable per al conjunt. Altres millores potencials serien l'increment del número de jugadors, els camps de la base de dades o les funcionalitats de la GUI.

2) *Interoperabilitat*: El disseny es basa en diferents dispositius que intercanvien informació. És necessari garantir el correcte funcionament en tot moment i considerar possibles substitucions en cas de necessitat. Els protocols de comunicació estàndards ajudaran a mantenir la connectivitat en el sistema.

3) *Modularitat*: L'arquitectura modular és necessària per a garantir la independència entre cadascun dels components i interfícies del sistema. Una bona definició de les APIs és essencial per a la correcta estructuració del codi i facilitar el procés de desenvolupament en col·laboració.

4) *Fiabilitat*: Les xarxes de comunicacions són sistemes especialment complexos d'implementar quan el temps real és un factor crític. És necessari garantir l'intercanvi de missatges de forma ràpida i precisa, considerant els casos pitjors amb interferències i pèrdua de paquets. Els protocols estàndards ofereixen mètodes específics (per exemple, reconeixements o retransmissió de paquets) per afegir robustesa a la transmissió. També és possible afegir fiabilitat extra a través de les capes superiors de software de l'stack de comunicacions.

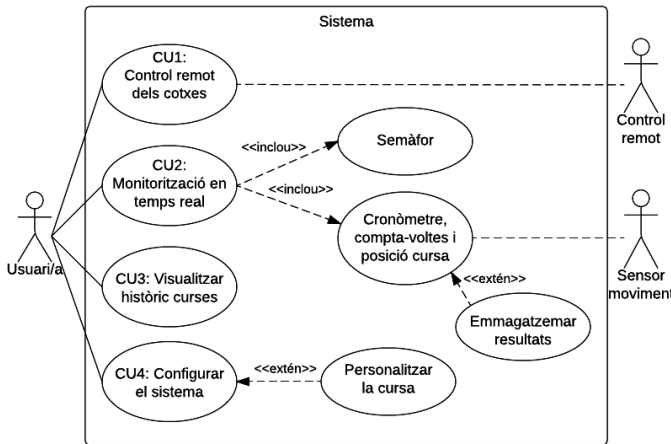


Fig. 1. Casos d'ús del sistema

5) *Consum energètic*: Un dels factors més crítics en el desenvolupament de xarxes de sensors és el consum d'energia i l'autonomia dels dispositius alimentats per bateries.

El disseny presenta una optimització d'energia basada en modes low-power i sleep en els nodes distribuïts. També s'ha donat una prioritat rellevant als protocols eficients i software dedicat a sistemes encastats a l'hora de seleccionar les tecnologies del disseny.

C. Requeriments funcionals

Les funcionalitats i comportament del sistema s'han definit a partir de quatre casos d'ús diferenciats, tal i com es mostra en la Figura 1.

1) *Cas d'ús 1: Control remot dels cotxes*: La potència de les pistes Slot s'ha de poder controlar a través d'un control remot. Hi ha d'haver un controlador independent per a cada pista.

2) *Cas d'ús 2: Monitorització en temps real*: La interfície gràfica ha de mostrar una pantalla d'interacció durant la cursa. Primerament, un semàfor indicarà el començament de la cursa. Seguidament, l'usuari/a ha de ser capaç de seguir la competició a través del cronòmetre, l'indicador de posició i el comptador de voltes.

De forma opcional es podran emmagatzemar els resultats de cada cursa en la base de dades.

3) *Cas d'ús 3: Visualització de l'històric de curses*: Les curses anteriors emmagatzemades s'han de poder mostrar a través de la GUI. Els camps a mostrar són *username*, el comptador de victòries, els resultats anteriors, informació dels circuits... L'històric de curses ha de ser accessible des de la interfície gràfica o bé la pàgina web.

4) *Cas d'ús 4: Configuració del sistema*: El sistema s'ha de poder configurar des de la GUI. Alguns paràmetres són el nombre de sensors de moviment, el número de voltes i el so i indicadors lumínics del sistema.

D. Arquitectura del sistema

Tal i com es mostra en la Figura 2, el sistema és gestionat des d'un node principal que es comunica amb dos tipus de perifèric: el control remot dels cotxes i els nodes de detecció per a la monitorització de la cursa.

1) *Dispositiu central*: Una de les característiques principals del sistema és la gestió de les dades, els recursos i la comunicació des d'un node exclusiu, i. e., el dispositiu central.

El seu disseny es basa en múltiples mòduls separats segons les tasques a realitzar. La correcta definició d'interfícies és essencial per a conservar-ne la independència, facilitar la integració amb mòduls addicionals o bé modificar els existents.

La interacció amb l'usuari es duu a terme a través de la interfície gràfica. El software de backend s'encarrega de la gestió de la cursa i la comunicació amb els perifèrics en temps real mentre que el frontend es dedica a la interacció amb l'usuari. Aquesta es basa en tres pantalles diferents:

- **Cursa**: La monitorització en temps real inclou semàfor, cronòmetre, comptador de voltes, indicador de posició... Ofereix també opcions per a iniciar o aturar la cursa així com emmagatzemar els resultats.
- **Històric**: Els resultats emmagatzemats es poden organitzar per Users (nombre de victòries, posició en el rànquing i resultats previs) o bé per Circuits (descripció i curses realitzades).
- **Configuració**: El sistema es pot configurar des d'una pantalla de la GUI.

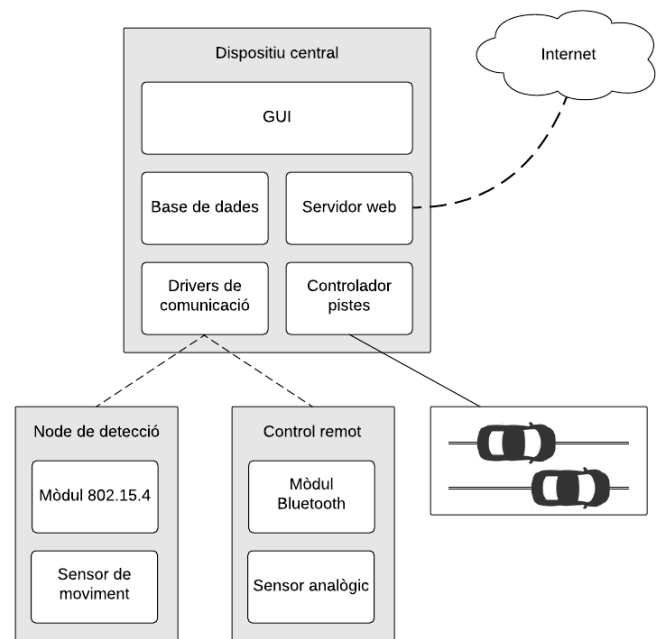


Fig. 2. Arquitectura del sistema

La informació s'emmagatzema en una base de dades al dispositiu central. Una característica essencial dels sistemes

IoT és que les dades emmagatzemades estiguin disponibles des d'internet. En aquest cas, el servidor mostra la informació de la base de dades en una pàgina web.

D'altra banda, el dispositiu central també s'encarrega de la comunicació amb els perifèrics. Tal i com es descriu en l'apartat anterior, la xarxa es basa en dos protocols de comunicació sense fils de baixa potència: BLE i 802.15.4. Per a garantir la comunicació es necessita un driver per a cada protocol.

Finalment, el dispositiu central compta amb un controlador per a la connexió amb la pista Slot.

2) *Control remot*: Tal i com s'explica anteriorment, la interacció amb els cotxes Slot ha de ser possible des d'un control remot. Aquest es basa en un mòdul BLE que envia contínuament informació de control al dispositiu central. Tot i que en la Secció IV es descriu una implementació específica, el disseny proposat no determina quin mètode utilitzar per al control. En el seu lloc, implementa una API oberta per a interactuar amb el dispositiu analògic desitjat. Per exemple, els cotxes es poden controlar mitjançant un potenciómetre connectat a una entrada analògica o bé a través de dispositius més complexos connectats al port SPI, I2C o sèrie.

3) *Node de detecció de moviment*: La monitorització en temps real és possible gràcies a diversos nodes de detecció de moviment localitzats al llarg de la pista Slot. El seu objectiu és detectar quan els cotxes travessen una posició determinada i informar al dispositiu central per a què actualitzi l'estat de la cursa en temps real.

El protocol 802.15.4 simplifica el muntatge facilitant la comunicació sense necessitat de cables físics. Els nodes es basen en un mòdul de comunicació amb un sensor de moviment connectat a una entrada analògica. El mòdul activa una interrupció a través de l'estímul del sensor i envia un missatge 802.15.4. També compta amb indicadors lumínics per a informar del començament i final de cursa.

Els nodes de detecció són gestionats des del dispositiu central segons la màquina d'estats de la Figura 3. El primer estat s'activa quan el dispositiu central busca nodes de detecció, de manera que aquests envien advertisements. Una vegada finalitzat el procés d'identificació de nodes, aquests entren a l'estat *idle* (amb el sensor i interrupcions desactivats). Al rebre el missatge *start*, realitzen un compta enrere (estat *countdown*) i entren a l'estat de *race*, enviant un missatge al detectar moviment. L'estat acaba quan el dispositiu central indica que algú ha finalitzat la cursa i els nodes entren a l'estat *winner*, activant els indicadors lluminosos del guanyador corresponent. Finalment, el procés torna a començar de nou saltant a *idle*.

III. IMPLEMENTACIÓ

Aquesta secció detalla la implementació del sistema proposat. El hardware descrit s'ha seleccionat seguint el següent criteri: cost, sistema obert, eficiència i fiabilitat. Tot i així, existeixen possibles alternatives per a cada component.

A. Dispositiu central: Raspberry Pi

La Raspberry Pi (RPI) és una plataforma de baix consum i baix cost que suporta múltiples sistemes operatius, incloent

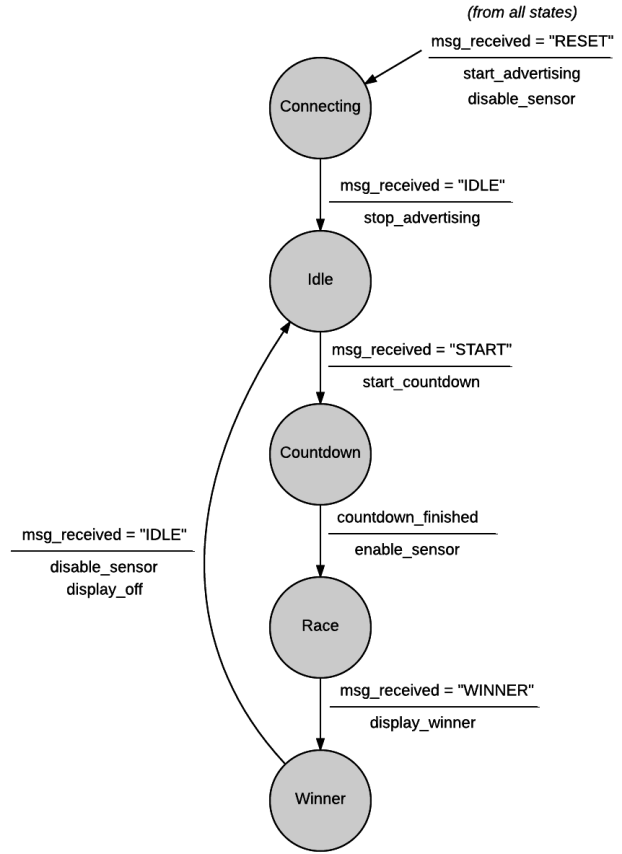


Fig. 3. Diagrama d'estats dels nodes de detecció

distribucions de Debian. El dispositiu ha obtingut recentment una gran popularitat en àmbits com la domòtica o l'educació [10] [14] [19] degut a la seva flexibilitat i facilitat d'ús. Ofereix diverses opcions de connectivitat com ara connexió Ethernet o USB.

La interfície d'interacció pot ser a través de connexió SSH o bé teclat, ratolí i una pantalla connectada al port HDMI. També ofereix pins d'entrada-sortida que permeten interrupcions, entrades analògiques, PWM, entre d'altres funcionalitats.

La RPi també ofereix facilitats de programació per a interactuar amb el sistema operatiu o hardware de forma simple. El llenguatge més popular és Python, proporcionant un gran nombre d'APIs i llibreries que utilitzen software més complex, així com mòduls del sistema.

El codi principal d'aquest sistema està escrit en Python, utilitzant llibreries específiques per a l'entorn gràfic, la temporització, la interacció amb els pins d'entrada-sortida o la comunicació sense fils amb els perifèrics, entre d'altres.

A banda del codi Python també s'utilitzen altres mòduls del sistema GNU/Linux, com ara el servidor Apache o la base de dades MySQL.

1) *Interfície gràfica*: Tal i com es descriu a la Secció III, la GUI s'encarrega tant de la interacció amb l'exterior com de la gestió de la cursa en temps real. La seva implementació es basa

en la divisió de software entre frontend i backend. El frontend utilitza la llibreria wxPython per a crear els gràfics. Aquesta està basada en wxWidgets, un software multi-plataforma que inclou components gràfics predissenyats (finestres, botons...).

2) *Base de dades i servidor web*: La informació és emmagatzemada en una base de dades MySQL implementada en el GNU/Linux de la Raspberry Pi. La base de dades està formada per quatre taules que es relacionen entre sí tal i com es mostra en la Figura 4. *Users* i *Circuits* emmagatzemen informació arbitrària per a personalitzar la partida, *Races* acumula els resultats de la competició (relacionant *Users* i *Circuits*) i *Laps* emmagatzema les voltes parcials de les curses de la taula *Races*.

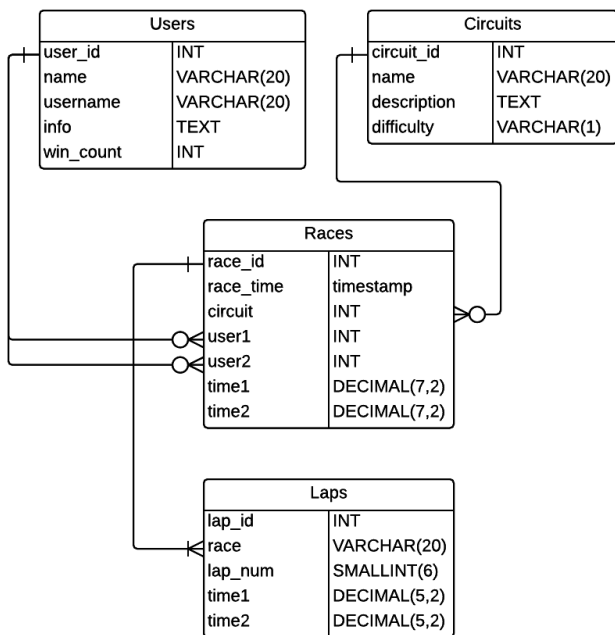


Fig. 4. Base de dades

Les dades són accessibles des d'internet gràcies al servidor web Apache. La implementació es basa en un script *cgi* que consulta la base de dades i en representa la informació en format *HTML*. Per a modificar l'estètica de la pàgina s'ha utilitzat el llenguatge *css*.

B. Control remot

A continuació es descriuen els components que intervenen en el sistema de control remot dels vehicles Slot.

1) *Sensor analògic. Acceleròmetre*: Tal i com es descriu en la Secció I, els mòduls de control remot ofereixen una API versàtil per a comunicar-se amb tot tipus de dispositius analògics. Aquesta es basa en funcions per a gestionar el perifèric (connexió i intercanvi de dades) que alhora utilitzen els drivers de baix nivell corresponents, i. e. comunicació SPI, I2C o bé Serial.

Aquesta implementació utilitza l'acceleròmetre de tres eixos *ADXL345* [20] per a controlar la potència de les pistes

a través de la variació de la inclinació del control remot. L'acceleròmetre té una resolució de 10 bits, als quals s'accedeix a través de transferències I2C.

El control es realitza a partir d'un sol eix, llegint els dos bytes corresponents de forma periòdica i enviant-los al dispositiu central a través del protocol BLE.

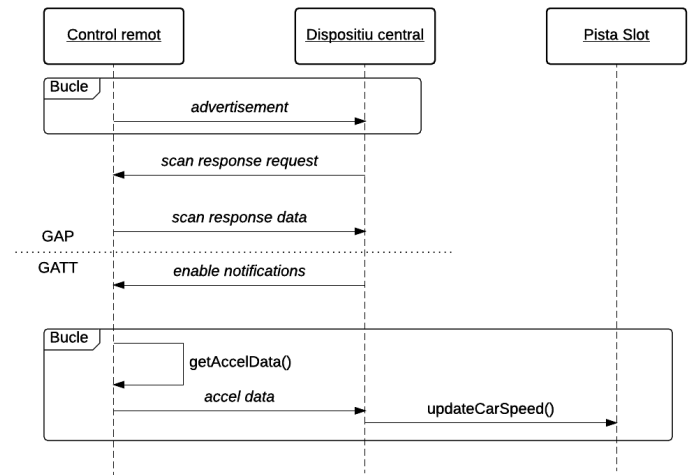


Fig. 6. Diagrama de seqüència del control remot

2) *Comunicació BLE*: En el protocol BLE els dispositius poden prendre dos rols: client i servidor. En aquest cas el dispositiu central actua com a client que envia peticions als servidors (mòduls de control remot).

Tal i com es mostra en la Figura 6, la comunicació es basa en dues fases: *Generic Access Profile (GAP)* a l'hora d'establir la connexió i *Generic Attribute Profile (GATT)* per a gestionar la comunicació una vegada els dispositius s'han associat.

El procés GAP es basa en l'intercanvi de tres missatges: primer els mòduls remots envien advertisements broadcast. El dispositiu central respon amb un scan response request i finalment els perifèrics confirmen la connexió amb un scan response data.

Una vegada els dispositius s'han associat, la comunicació és gestionada des del dispositiu central mitjançant transaccions GATT. El protocol BLE presenta un mecanisme de notificacions que permet al perifèric prendre la iniciativa i enviar dades sense la petició del client. D'aquesta manera, una vegada el client habilita les notificacions, el mòdul de control remot envia les dades de l'acceleròmetre periòdicament. El dispositiu central processa els missatges i actualitza la potència dels cotxes.

La implementació dels mòduls de control remot es basa en la plataforma integrada *Radino nRF8001*. Aquesta es compon per un microcontrolador *Atmel ATmega32U4* i un transceptor BLE *nRF8001*. El codi s'ha implementat al microcontrolador utilitzant els drivers BLE específics que proporciona l'empresa desenvolupadora. El microcontrolador compta amb una interfície USB per a la programació i el port sèrie i una interfície I2C per a comunicar-se amb l'acceleròmetre.

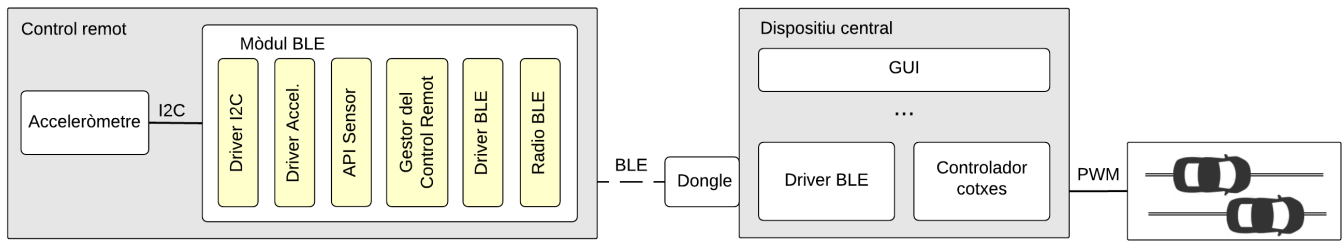


Fig. 5. Diagrama de blocs del control remot dels vehicles

3) *Controlador PWM*: Per a controlar els cotxes Slot en el circuit cal general una tensió variable entre 0 i 20 V. La Figura 7, mostra un mètode simple per a crear una tensió regulable a través de la tècnica de modulació per amplada de pols, de l'anglès Pulse Width Modulation (PWM). El PWM es basa en variar el rendiment de cicle d'un senyal per a obtenir una tensió contínua al nivell desitjat. Com que la tensió màxima dels pins de sortida de la RPi està limitada a 3,3 V, s'ha afegit un transistor que commuta el circuit de potència (20 V) a la freqüència de PWM. La implementació utilitza un mòdul específic del Kernel dedicat al PWM juntament amb la corresponent llibreria de Python. Aquesta ofereix funcions per a activar, desactivar i modificar el rendiment de cicle del pin d'entrada-sortida desitjat.

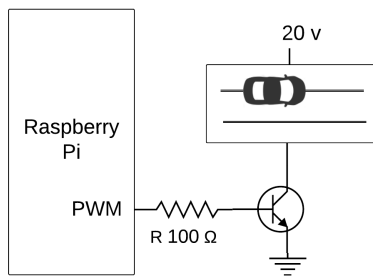


Fig. 7. Circuit de control per a un cotxe

C. Monitorització de curses en temps real

El sistema de monitorització inclou el dispositiu central i els nodes de detecció de moviment.

1) *Gestió de la cursa. Mòduls Python*: Tal i com s'indica anteriorment, la gestió de la cursa s'ha implementat com a backend de la interfície gràfica. S'ha utilitzat exclusivament el llenguatge Python, estructurant el codi segons el diagrama de la Figura 8. Les classes resultants són les següents:

- Slot: Menú principal de navegació
- Race: Programa principal que gestiona la cursa
- Semaphore: Implementació del semàfor per a començar la cursa
- PreviousData: Programa que gestiona la visualització de l'històric de curses
- Users: Classe creada per a emmagatzemar la informació dels usuaris
- Races: Classe creada per a emmagatzemar la informació de les curses

- DbConnection: Classe que ofereix les funcionalitats bàsiques per a comunicar-se amb la base de dades

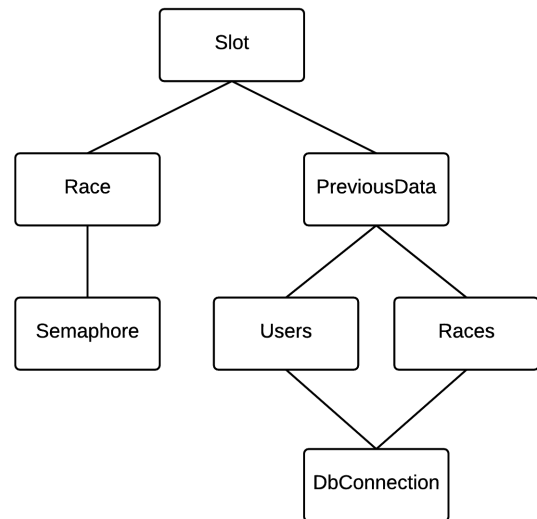


Fig. 8. Diagrama de classes

2) *Nodes de detecció de moviment*: Els nodes de detecció de moviment es troben distribuïts al llarg del circuit Slot i s'encarreguen d'avisar al dispositiu central quan detecten moviment en un punt determinat de la pista. Encara que el nombre de nodes a utilitzar és variable i configurable, en aquest prototip se n'han utilitzat dos. Els nodes es basen en un mòdul AVR-ZigBit, que compta amb un microcontrolador Atmel ATmega1281 i un transceptor 802.15.4. El microcontrolador implementa la màquina d'estats de la Figura 3, utilitzant el sistema operatiu Contiki. Això és possible ja que AVR-ZigBit és una de les plataformes per a les quals existeix la portabilitat de Contiki.

3) *Contiki. Rime*: Contiki té múltiples aplicacions implementades per defecte, així com exemples demostratius per a utilitzar funcionalitats del sistema. Tal i com s'ha introduït en la Secció II, també compta amb un stack de comunicacions TCP/IP anomenat uIP. Tanmateix, paral·lelament ofereix una alternativa més simple i lleugera de protocol de comunicacions: Rime [21]. Rime es basa en un seguit de primitives de comunicacions bàsiques per tal de garantir funcionalitats similars a uIP. Per exemple, la funció més bàsica de Rime és la comunicació broadcast sense identificació de cap tipus,

anomenada abc. Mentre que per a transmetre informació a un node concret s'utilitza el mode unicast.

IV. RESULTATS

Aquest article presenta el disseny i implementació d'un sistema de control i monitorització de curses Slot basat en un dispositiu central i una xarxa de perifèrics que es comuniquen a través dels protocols 802.15.4 i BLE. Per tal de validar la proposta i testejar el sistema en un entorn real de Slot, s'ha creat un prototip. En aquesta secció es detalla el sistema resultant així com l'avaluació dels aspectes crítics més representatius.

A. Prototip d'implementació

Tal i com s'explica en la Secció II, el sistema es centralitza en una Raspberry Pi, utilitzant-ne el següent hardware:

- Port HDMI: La interfície gràfica es basa en una pantalla HDMI. La interacció amb aquesta es realitza a través de teclat i ratolí connectats al port USB
- Connexió Ethernet: La connexió amb la xarxa (tant local com exterior) es duu a terme a través del port Ethernet
- Comunicació BLE: Un *dongle* Bluetooth connectat al port USB i les llibreries específiques per a GNU/Linux han permès la comunicació amb els perifèrics BLE
- Pins d'entrada-sortida: Els pins d'entrada-sortida actuen com a circuit de potència de les pistes (PWM)
- Port Sèrie: La comunicació sèrie s'ha utilitzat com a alternativa a la pantalla HDMI per a debugar el sistema
- Sortida d'àudio: La connexió d'àudio pròpia de la plataforma ha permès la reproducció de sons per a complementar les partides Slot

B. Avaluació del prototip

A continuació es descriuen els aspectes que s'han tingut en compte a l'hora d'avaluar el sistema.

1) *Monitorització en temps real*: Per a garantir la monitorització en temps real cal tenir en compte dos factors crítics. D'una banda cal garantir que la temporització del cronòmetre de la RPi presenta la precisió necessària. Això és possible ja que des del programa Python es crida al rellotge del sistema (250 MHz) cada vegada que hi ha un esdeveniment. Pel que fa al refresc del cronòmetre es realitza a una freqüència de 10 Hz, mostrant per pantalla les unitats de fins a dècimes de segon.

D'altra banda, cal tenir en compte que a l'hora de monitoritzar la cursa hi intervenen els nodes de detecció de moviment col·locats al llarg de la pista. És per tant important garantir que el retard en l'enviament de missatges no és significatiu. Tal i com es mostra en l'equació 1, la latència teòrica que ofereix el protocol 802.15.4 depen de la mida del paquet [22]. Per a paquets de 23 bytes (mínim utilitzant adreçament), cal un temps de $4110\mu s$, en el millor cas. Aquest retard no ha de suposar un inconvenient ja que el sistema treballa amb resolució de centèsima.

$$T_{total} = P_{mida} * (2 * 65) + 32 * (P_{mida} + 12)\mu s \quad (1)$$

Encara que s'hagi comprovat un correcte comportament de la monitorització durant les curses de prova, s'ha de tenir en compte que el temps real és un factor crític i caldria estudiar el sistema en major precisió, exercici que s'ha deixat per a línies futures del projecte.

2) *Comunicació sense fils*: Un altre dels factors crítics que intervenen en el disseny exposat és la fiabilitat de la comunicació sense fils. En les especificacions dels protocols es detallen les seves prestacions tècniques i els mecanismes amb què compten per a garantir la fiabilitat. Tot i així, a la realitat poden presentar-se casos concrets en què els protocols no es comportin de forma desitjada. Una de les complicacions amb què compta el disseny és la coexistència dels protocols 802.15.4 i Bluetooth, els quals treballen en la mateixa banda de freqüència (2.4GHz).

No s'han detectat pèrdues de paquets ni intermitències en la connexió BLE durant les proves realitzades, segurament degut a què el nombre de missatges 802.15.4 enviats és reduït en comparació a la transmissió BLE contínua.

Cal tenir en compte que el sistema s'ha testejat en un entorn òptim en el qual no hi havia altres dispositius de comunicació sense fils en el mateix rang. Un experiment interessant que s'ha deixat com a línia de treball futura seria comprovar si el sistema continua essent robust amb la presència d'interferències externes, per exemple, una xarxa Wi-Fi.

C. Llicència de programari

Com s'ha comentat en diverses ocasions, tant el hardware com el software d'aquesta proposta són oberts. També ho és la implementació del sistema final, el codi del qual s'ha penjat a la plataforma de distribució *GitHub* sota llicència GPLv2. S'ha escollit aquesta llicència amb *copyleft* per tal de facilitar el seguiment de l'evolució del projecte i garantir que les possibles modificacions siguin també lliures.

V. CONCLUSIONS

Aquest article representa un àlisis de la tecnologia utilitzada en les curses Slot, íntimament relacionada amb les xarxes de comunicacions d'IoT. Havent detectat la falta de solucions lliures i obertes en el mercat, s'ha proposat el disseny d'un sistema que unifica les funcionalitats essencials per a monitoritzar i controlar curses Slot en una plataforma oberta. El disseny es basa en els protocols de comunicació de baixa potència 802.15.4 i BLE.

Com a línies de treball futures s'han deixat els tests en major profunditat, considerant casos crítics amb interferències en la comunicació i estudiant la temporització del sistema de forma precisa. També caldrà realitzar un seguiment de l'evolució del projecte en cas de possibles col·laboracions en comunitat.

BIBLIOGRAFIA

- [1] L. Tan and N. Wang, "Future internet: The internet of things," in *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, vol. 5, 2010, pp. 376–380.
- [2] N.-M. S. Jaeseok Yun, Il-Yeop Ahn and J. Kim, "A device software platform for consumer electronics based on the internet of things," *IEEE Transactions on Consumer Electronics*, vol. 61, no. 4, pp. 564–571, Nov. 2015.

- [3] J. N. Jinsung Byun, Bounju Jeon and S. Park, "An intelligent self-adjusting sensor for smart home services based on zigbee communications," *IEEE Transactions on Consumer Electronics*, vol. 58, pp. 794–802, 2012.
- [4] E. Z. Adriana Wilde, Richard Oliver, "Developing a low-cost general-purpose device for the internet of things," in *Sensing Technology (ICST), 2013 Seventh International Conference on*, 2013, pp. 490–494.
- [5] Akber. Slot car history. [Online]. Available: <http://www.slotcarhistory.com/>
- [6] Slot, más que un juego de coches una historia de modelismo a escala. [Online]. Available: <http://slotrc.com/posts/2013/12/26/modelismo-coches-slot-historia/>
- [7] Scs slot car wireless controllers for scalextric sss and scx-d. [Online]. Available: <http://www.slotcarsolutions.com/>
- [8] Real fx could be the r/c slot car racing of the future. [Online]. Available: <http://www.autoblog.com/2014/06/24/real-fx-rc-slot-car-racing-future-video/>
- [9] Camaro slot car carrera. [Online]. Available: <http://www.asciimation.co.nz/camercar/>
- [10] A. A. A. Sudhir G. Nikhade, "Wireless sensor network communication terminal based on embedded linux and xbee," in *Circuit, Power and Computing Technologies (ICCPCT), 2014 International Conference on*, 2014, pp. 1468–1473.
- [11] C.-C. S. Jin-Shyan Lee, Yu-Wei Su, "A comparative study of wireless protocols: Bluetooth, uwb, zigbee, and wi-fi," in *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE, 2007*, pp. 46–51.
- [12] Y.-H. S. Jin-Shyan Lee, Ming-Feng Dong, "A preliminary study of low power wireless technologies: Zigbee and bluetooth low energy," in *Industrial Electronics and Applications (ICIEA), 2015 IEEE 10th Conference on*, 2015, pp. 135–139.
- [13] C. C. Robert Davidson, Akiba and K. Townsend, *Getting started with Bluetooth Low Energy*. O'Reilly Media, Inc., 2014.
- [14] S. G. Nikhade, "Wireless sensor network system using raspberry pi and zigbee for environmental monitoring applications," in *Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), 2015 International Conference on*, 2015, pp. 376–381.
- [15] W. L. Xin Ma, "The analysis of 6lowpan technology," in *Computational Intelligence and Industrial Application, 2008. PACIIA '08. Pacific-Asia Workshop on*, vol. 1, 2008, pp. 963–966.
- [16] T. N. H. Thang Vu Chien, Hung Nguyen Chan, "A comparative study on operating system for wireless sensor networks," in *Advanced Computer Science and Information System (ICACSIS), 2011 International Conference on*, 2011, pp. 73–78.
- [17] A. Dunkels, "Contiki - a lightweight and flexible operating system for tiny networked sensors," in *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, Nov 2004.
- [18] I. S. onela Halcu, Grigore Stamatescu and V. Sgarciu, "An analysis of security and communication constraints of ipv6-based sensor networks," in *Electronics, Computers and Artificial Intelligence (ECAI), 2014 6th International Conference on*, Oct. 2014, pp. 55–60.
- [19] P. Sachdevaa and S. Katchii, "A review paper on raspberry pi," *International Journal of Current Engineering and Technology*, Nov. 2014.
- [20] S. Fun. 3-axis digital accelerometer adxl345 datasheet. [Online]. Available: <http://www.sparkfun.com/datasheets/Sensors/Accelerometer/ADXL345.pdf>
- [21] A. Dunkels, "Rime - a lightweight layered communication stack for sensor networks," in *European Conference on Wireless Sensor Networks (EWSN)*, January 2007, pp. 455–462.
- [22] J. Delsing, J. Eliasson, and V. Leijon, "Latency and packet loss of an interfered 802.15.4 channel in an industrial environment," in *International Conference on Sensor Technologies and Application*, July. 2010, pp. 33–38.