

PAC 7

ENTREGA FINAL DEL PROGRAMARI

Nom: Germán Almudever Larrodé

TFG – Sistemes Encastrats

Consultor: Jordi Bécares Ferrés

Grau de Tecnologies de les Telecomunicacions 2015-16

Tasques realitzades

◦ Quins punts s'han finalitzat dels objectius marcats en la planificació

Aquests són els objectius assolits amb l'elaboració del codi final:

- Integrar els detectors d'intrusió (Contacte Magnètic i Detector Infraroig Passiu) al sistema. La connexió dels sensors amb la placa microprocessadora és directa, no cal circuiteria adicional. La gestió d'aquest HW la implementa el mòdul *SENSORS*, que depèn dels drivers *GPIO* i *GPIOINT*, i actua com a *middleware* (és un mòdul específic de la aplicació de Alarma) entre l'aplicació i els mòduls genèrics (drivers).
- Afegir dos indicadors d'acció (LEDs) per informar de l'estat del sistema: un LED d'estat (verd) i un de connexió (vermell).

Igual que en el cas dels sensors, no cal adaptació de HW per integrarlos. El control del mateixos s'implementa en el mòdul *LEDCTRL*. Aquest executarà una tasca (FreeRTOS task) que actualitza l'estat del LEDs. Estat que es podrà configurar mitjançant al una API senzilla que abstreu dels detalls de HW.

- Incorporar una interfície d'usuari al sistema (keypad matricial de 3x4 tecles). S'utilitzen 7 pins de la placa per connectar el teclat matricial. L'scan del teclat el realitza el mòdul *KEYPAD* (driver), i les pulsacions de tecles que aquest recull alimenten un mòdul de més alt nivell, *CMD* que sap traduir seqüències de tecles a comandes individuals. Aquest sistema de gestió del teclat i reconeixement de comandes s'executa en una tasca autònoma (*task_cmd*), i quan una comanda és recoeguda, es comunica amb la tasca principal (*task_fsm*), que és qui reacciona, com correspongi, a la comanda.
- Oferir dues modalitats d'armat del sistema (total i parcial). El funcionament centralitzat i la gestió de les comandes d'usuari, inputs dels sensors, i activació de LEDs, o el *SISCOM* (detallat més avall) es troba implementat en 2 mòduls, específics per a l'aplicació, *FSM* i *FSMSTATE*, que gestionen la màquina d'estat que controla els diferents estats del sistema.
- Dotar el sistema de connexió GPRS per enviar notificacions del sistema via SMS. Els detalls de la comunicació pròpiament dita estan gestionats per un equip *SISCOM* (sistema de comunicació GSM/GPRS bidireccional). La aplicació fa servir el mòdul *SISCOM* per controlar aquest equip i enviar les notificacions en funció de l'estat del sistema.

- Permetre el monitoratge de les accions realitzades via UART. Totes les accions que es generen, ja sigui en l'execució del programa com a l'aplicació del test, es donen de sortida pel port sèrie, que es pot verificar mitjançant la connexió d'una aplicació de terminal (per exemple, PuTTY).

◦ Quins objectius no s'han complert?

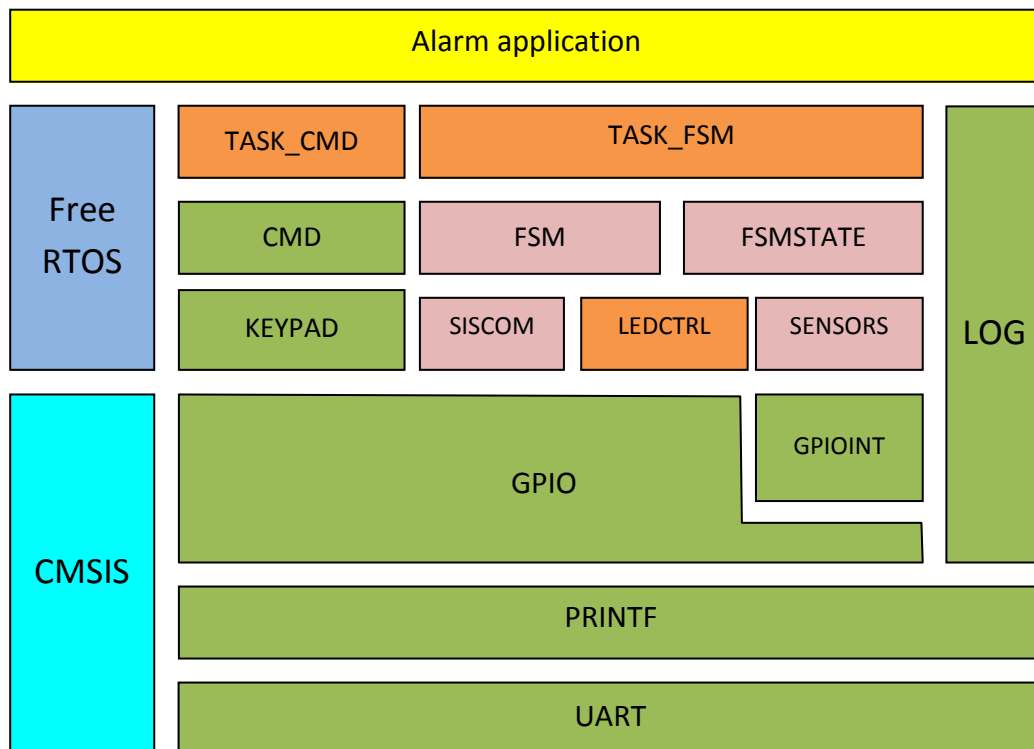
Tots els objectius principals s'han assolit satisfactòriament. S'ha de dir que durant la realització de les PACs a la primera fase no es van completar amb èxit la *PAC4 – Driver WiFly* (no va funcionar totalment) ni la *PAC5 – Productor Consumidor* (per manca de temps al estar encallat amb la *PAC4*). Amb aquest motiu, i veien que podia retardar i comprometre l'èxit del projecte, es va decidir juntament amb el consell del consultor no incorporar la comunicació amb el WiFly fins no disposar del funcionament propi del sistema de seguretat complert.

La comunicació amb Internet es realitza mitjançant el mòdul de HW WiFly que es connecta a la UART. Els drivers *ARPALAB* i *WIFLY* es troben totalment implementats però sense funcionar d'una manera fiable. De fet, s'ha reimplementat el mòdul UART proporcionat, extenent la seva funcionalitat per suportar modes amb i sense interrupcions.

Els problema que m'he trobat era que no s'establí connexió al executar la comanda "join": Fallava la autenticació a l'hora de la connexió Wi-Fi. S'han realitzat tot tipus de proves de xifratge (WPA, WEP, etc...inclòs accés obert) sense resultat. Els mòduls es troben incorporats a la llibreria “*encastats*” per futures millores del sistema.

Respecte a la funcionalitat que es demanava a la *PAC5*, ha calgut implementar un mecanisme amb processos productors i consumidors -amb la corresponent sincronització via mutex- per a comunicar els diferents subsistemes de la aplicació.

◦ Diagrama de blocs del sistema



◦ Sistemes de test fets servir

S'ha implementat un projecte de test, i un mòdul específic, TEST que proporciona el framework mínim sobre el que es pot executar una sèrie de tests (test suite). Aquests test permeten efectuar comprovacions de diverses menes.

Hi ha tests manuals, que precisen la interacció i/o la validació per part de l'usuari, i tests automàtics, que s'executen desatesos i donen un resultat binari respecte a la validació d'un component, mòdul o funcionalitat. També es poden emprar els tests per validar la integritat i el correcte funcionament del maquinari.

Autoavaluació

◦ Explica quin ha sigut el punt més complicat de la PAC i com l'has resolt

El punt més complicat i que ha condicionat el desenvolupament del sistema ha estat la resolució dels mòduls *ARPALAB* i *WIFLY*, que malauradament no s'han pogut aprofitar per no establir una connexió fiable amb la xarxa sense fils.

Conèixer el maquinari i la seva gestió també ha estat laboriós. Aprendre la programació dels diferents registres que configuren cada funcionalitat disponible del LPC1769 ha requerit força temps. D'altra banda, el sistema FreeRTOS es troba ben documentat i es troben molt enllaços d'ajuda i consulta.

També, aprendre a manegar el LPCXpresso i gestionar 4 projectes que tenien fitxers comuns ha estat un bon exercici.

◦ **Exposa (si n'hi ha) els bugs i errors en el sistema desenvolupat**

Degut als problemes ja mencionats amb el mòdul de *WIFLY*, s'ha fet una re-implementació del mòdul *UART*, doncs es sospitava de limitacions en la recepció. Al final, es va optar per reimplementar-lo, i fins i tot, per donar suport a un mode sense interrupcions, a on la recepció era síncrona i bloquejant. Configuracions que es poden establir per separat per a cada una de les UARTS (es continua sense suportar la UART2).

En el procés, s'ha trobat que la UART1 no funcionava bé. En particular, en inicialització de la UART1 -mode amb interrupcions-, s'ha canviat

```
LPC_PINCON->PINSEL4 &= ~0x0000000F;
LPC_PINCON->PINSEL4 |= 0x0000000A; /* Enable RxD1 P2.1, TxD1 P2.0 */
```

per el codi següent:

```
//Enable TxD1 (pin P0.15)
LPC_PINCON->PINSEL0 &= ~0xC0000000; //P0.15 is bits 31:30
LPC_PINCON->PINSEL0 |= 0x40000000; //TxD1 is 01
//Enable RxD1 (pin P0.16)
LPC_PINCON->PINSEL1 &= ~0x00000003; //P0.16 is bits 1:0
LPC_PINCON->PINSEL1 |= 0x00000001; //RxD1 is 01
```

És possible que el codi original fos bo per a la versió LPC1700 (és el que figura en la capçalera del mòdul UART proporcionat).

Ambdues versions s'han mantingut al codi, per a poder comparar i estudiar el problema amb detall. Vegeu el "#if 0..." en la funció *UART_init_with_interrupts()*.

S'han trobat varietat de bugs menors durant el desenvolupament, que s'han pogut aïllar i corregir gràcies als tests.

Veureu que hi ha diversos tests per a qüestions que no son relatives al HW ni de l'àmbit dels sistemes encastats, sinó que són propis de la programació en general. Això és així per que els test s'han anat fent sota demanda.

Sovint, depurar el propis tests ha estat determinant per a trobar el problema.

Un aspecte que cal remarcar és que tot el manegament de memòria s'ha fet estàticament: no es fa servir **malloc()** ni cap funció de reserva dinàmica de memòria en tot el codi. El motiu era eliminar la fragmentació de memòria. I inicialment, no estavem segurs de quines interaccions es podrien donar al fer servir interrupcions i memòria dinàmica.

◦ **Warnings controlats i no en el workspace**

No hi ha cap “*warning*” al codi. Tots els projectes s'han compilat amb el setting “-Wall” (tots els warning activats), i absolutament tots els warnings emesos pel compilador s'han anat eliminant.

Degut a un possible bug del compilador,

(https://gcc.gnu.org/bugzilla/show_bug.cgi?id=53119), sovint apareixia el warning

“*missing braces around initializer [-Wmissing-braces]*”

en zero-initializations, per exemple, d'arrays d'estructures:

```
Foo foo[3] = { 0 };
```

després d'una mica d'investigació, s'ha pogut eliminar aquest warning amb la sintaxi

```
Foo foo[3] = {{ 0 }};
```

◦ **Temps total dedicat (aproximadament)**

La durada total per la realització del codi final segons la planificació era de 24 dies naturals. S'ha de tenir en compte que les hores disponibles són limitades degut a les obligacions professionals i familiars. Tot i això, es realitza l'entrega del codi es realitza sense exhaurir la data límit.

El total d'hores treballades és d'unes 100 hores. Altrament, ja es disposava de bona part del codi realitzat a les PAC de la primera fase.

Conclusions

En general, el treball realitzat ha estat satisfactori. S'ha implementat un sistema de seguretat complet, que compleix amb totes les funcionalitats designades als objectius, i es totalment operatiu.

S'ha dedicat força esforç implementar els tests i el framework de test, cosa que ha permès fer un desenvolupament progressiu i iteratiu.

També ha contribuït, en el mateix sentit, la decisió de crear mòduls que no eren “drivers”, sinó mòduls d'aplicació, (SISCOM, SENSORS, FSM...), que han estat provables per separat, i s'ha pogut refinar la interface dels mateixos i explorar variants arquitecturals.