

Servei per controlar sistemes encastrats

PAC 7 – Entrega final del programari

E.T. Informàtica de Sistemes

Estudiant
David Otaol Esclarín

Data de lliurament 10 de Juny de 2015

1. Tasques realitzades

Punts finalitzats dels objectiu marcats en la planificació.

He finalitzat els següents punts:

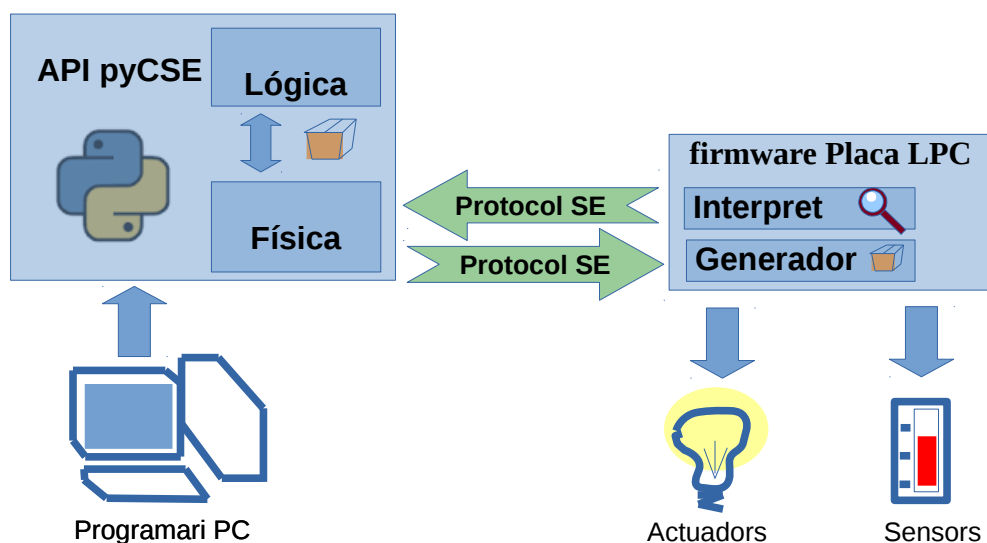
- Programar i verificar el funcionament de la API amb el port Serie.
- Programació bàsica de la aplicació per cridar a la API i veure els resultats en mode Text.
- Programació i implementació del protocol en la API.
- Programació i implementació del protocol en el FirmWare.
- Desenvolupar funcions en el FirmWare per treballar amb el Led.
- Desenvolupar funcions en el FirmWare per treballar amb el ADC.
- Desenvolupar funcions en el FirmWare per treballar amb el GPIO.
- Desenvolupar funcions en el FirmWare per treballar amb la UART.
- Desenvolupament de les funcions de la API per fer servir, via protocol, les funcions del FirmWare.
- Programació en el FirmWare de tasques de execució periòdica.

Punts no finalitzats de la planificació.

No he pogut finalitzar els següents punts:

- Desenvolupar funcions en el FirmWare per treballar amb I2C.
- Desenvolupar funcions en el FirmWare per treballar amb SPI.
- Desenvolupament de aplicació per representar gràficament les dades del sensor.

Diagrama de Blocs del Sistema.



Sistemes de Test fets Servir

He realitzat les següents proves a mesura que avançava el desenvolupament.

Fases de proves en ordre cronològic:

1. He provat la connexió per el port serie, llegint i enviant missatges per la UART3, amb les funcions Printf_print() i Printf_Scan()
2. Per provar el funcionament de la API, i he fet la aplicació en mode text, en Python.
3. He s per provar el funcionament de enviament de sol·licituds amb el protocol, he realitzat enviament de sol·licituds de encendre i apagar el Led.
4. He provat el funcionament de la recepció de dades amb el protocol, rebent dades del ADC.
5. Per provar la execució de tasques periòdiques, he fet proves de temporització del Led.
6. Per provar la recepció de multiples paquets amb el protocol he fet proves de temporització amb el ADC.
7. Per provar el GPIO s'ha fet servir el pin 24, que correspon al Led (encendre, apagar i veure estat).

Amb aquestes fases de proves, simultaneament s'han provat els diferents controls dels diferents dispositius.

2. Autoavaluació

Complexitat

Els punt mes complicat s de la Pac han sigut:

1. Programa els moduls de control del ADC, GPIO, SPI, etc. requereix coneixement tècnics avançats, i es massa complex. Per això he fet servir mòduls de exemple de la pàgina del fabricant del processador Cortex: <http://www.brc-electronics.nl/librarys>
2. La generació de seqüències de dades amb els format definits a sigut problemàtica.

El motiu es que tant en Python com en C, al definir per exemple una estructura amb el següent format :

Numèric		Numèric	Altres dades (Inici byte 4)
2 bytes		1 byte	n bytes
x	x	y	stream

Per motius de optimització (per treballar amb paraules de 4 bytes), tant C com Python la seqüència de bytes que generen es:

Numèric		Numèric		Altres dades (Inici byte 5)
2 bytes		2 bytes		n bytes
x	x	0	y	stream

I per tant s'hem produïen desplaçaments de bytes que m'impedien reconèixer correctament les dades.

Per resoldre-ho, en Python he de treballar amb estructures indicant el signe de format « > ».

Per resoldre-ho en C, he tractat totes les dades a nivell de Bytes.

3. Mida Variable.

Al ser paquets de mida variable, la localització i recepció de multiples respostes per lectures seqüencials, es complica una mica. Perquè si es llegeixen blocs de mida fixa de la UART, poden quedar el paquets tallats en dos blocs. En Python ho he resolt depurant una mica el codi,

4. Little Indian i Big Indian.

He tingut problemes de sincronització de format. Les dades generades en Python s'envien en Big Indian, però després el C els treballa en little Indian. Per tant he hagut de fer inversió del ordre dels bytes, en alguns punts, per fer comparacions de números i identificadors. Les dades numèriques de les respostes també les envia C en little Indian, i el Python les recull com a tal.

5. Interpretació de dades del sensor.

Les dades del sensor del ADC venen en tics, de 12 bits (segons el manual del LPC1769), i s'han de convertir. Per això es multipliquen per el voltatge (3,3 volts) i es divideixen per 4096 (2^{12}).

6. Printf sense strcpy.

Al principi els missatges del protocol s'hem quedaven tallats, ja que feia servir les funcions Printf_print que contenien strcpy(), que només copia fins trobar un 0 binari. Ho he arreglat treient el strcpy() de la funció Printf_scan, i creant una funció nova de enviament de bytes a la UART Printf_stream() sense strcpy.

Bugs i errors en el sistema desenvolupat

Per falta de temps no he a la aplicació cap tipus de validació de que les dades introduïdes siguin correctes, per la qual cosa, si es introdueixen dades incorrectes es produeixen diferents errors.

Després de recollir dades del ADC de forma periòdica, la placa el penja per un Handle error de la tasca, i deixa de funcionar fins que es reinicia. No he tingut temps de determinar, el motiu.

Volia donar la possibilitat de fer servir la funció Log, per possibilitar el codi, però l'he tret perquè de vegades es penja, i no m'ha donat temps de determinar el motiu. En canvi totes les funcions del modul Printf hem funcionen bè.

Hi han parts del projecte que no he pogut provar per manca de previsió i de temps, com la UART, el SPI, el I2C, i que per tent probablement no funcionin correctament.

Warnings controlats i no en el workspace

No he deixat cap Warning a la part del LPCXpresso, ni a la banda del Python.

Temps total dedicat (aproximadament)

Temps total aproximat dedicat al projecte 115 hores, sense tenir en compte el treball previ de les Pacs.

3. Autoavaluació

Conclusions: S'han complert els objectius?

Els objectius s'han complert parcialment.

He conseguit desenvolupar algunes de les parts, però altres no m'ha donat temps.

Tot i que he assolit varies de les funcionalitats bàsiques, moltes altres no m'ha donat temps de desenvolupar-les o acabar de depurar-les.

A mes hem dono conte que per esdevenir una eina potent i útil, faltaria ampliar i millorar molts aspectes del codi i de funcionalitats. Per exemple donar possibilitat de fer servir el wifly, afegir paràmetres al protocol per indicar si les dades venen en little Indian o Big Indian. Etc.