# Pràctica 6: Sistema de control per a un ascensor (III).

**Programació Concurrent i en Temps Real — iTIC**

Antoni Escobet Canal

5 de novembre de 2021

## Índex



## 1 Organització

Aquesta sessió continua amb l'objectiu de dissenyar i implementar un sistema de control per a una màquina elevadora (ascensor).

Fins aquest moment el sistema que s'ha construït és completament programari. Això pot induir a creure que `Erlang` és un llenguatge interessant com a tal però que no és vàlid quan es vol aplicar a sistemes reals en que cal interactuar amb el maquinari. Res més lluny de la realitat: recordeu que `Erlang` va ser dissenyat precisament per interactuar amb maquinari.

Aquesta pràctica té com a objectiu experimentar una de les estratègies per connectar una aplicació `Erlang` amb maquinari específic. Això es farà modificant la pràctica 5 per tal que la botonera de cabina deixi de ser simulada en la pantalla i passi a ser una botonera real. A tal efecte caldrà substituir el mòdul `bcab` per un altre mòdul funcionalment equivalent però que només fa de proxy amb la botonera física. Per aquest objectiu, no cal modificar res més de la pràctica 5.

Per dur a terme aquesta pràctica cal l'Arduino i un *shield* específic que inclou una botonera figura 1. Aquest *shield* s'us subministrarà el dia del control de pràctiques. El podreu tenir mentre dura la pràctica, i caldrà tornar-lo en acabar-la.

Amb l'objectiu de reforçar l'hàbit d'usar sistemes de control de versions, cal desenvolupar la pràctica amb el suport del sistema que ofereix `http://escriny3.epsem.upc.edu`.

Al contrari de les anteriors en aquesta pràctica no es dona el disseny explícit del programari. Se'n dona l'especificació funcional i es deixa el disseny a les vostres mans. Cal parar compte amb aquest punt atès que requereix un esforç previ de disseny de la solució que no requerien pràctiques anteriors.

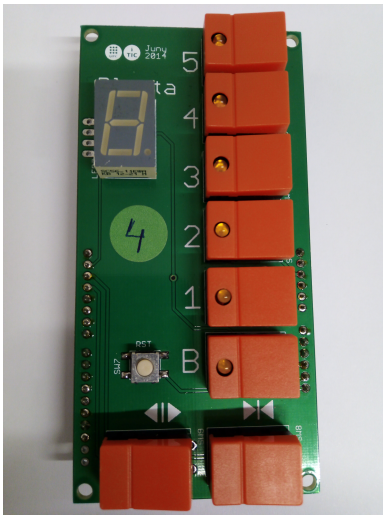


Figura 1: Fotografia de la *shield* BCAB.

### 1.1 Lliurament

Cal lliurar els exercicis en un tarfile a través d'Atenea en la data fixada. Cal que el desenvolupament es faci usant `Subversion` a través de les facilitats que ofereix `http://escriny3.epsem.upc.edu`. Al mateix temps caldrà presentar oralment la pràctica durant la classe de laboratori que ja s'anunciarà. La durada estimada d'aquesta pràctica és de 3 setmanes.

## 2 Shield BCAB

El *shield* BCAB per a l'Arduino és una placa dissenyada per l'EMIT que incorpora, entre altres elements, un conjunt de sis polsadors de pis, dos polsadors per obrir i tancar la porta, sis leds de pis i un visualitzador de 7 segments. L'esquema de la placa el podeu veure a la figura 2. En aquesta pràctica usarem aquesta placa com si es tractés d'una botonera d'ascensor amb polsadors retroil·luminables.

## 3 Desenvolupament

### 3.1 Aplicació a l'Arduino

A l'Arduino cal implantar una aplicació que condueix els leds i polsadors comandada per un protocol a través del port sèrie. L'aplicació cal implementar-la amb els recursos de les llibreries que heu anat dissenyant i implementant en assignatures anteriors.

El protocol que es parla a través del port sèrie és el següent (des del punt de vista de l'Arduino):

**rep 'Ex'** Si rep els caràcters `'Ex'` on `x` pot ser 0,1,2,3,4 o 5, encén el led corresponent (si estava apagat).

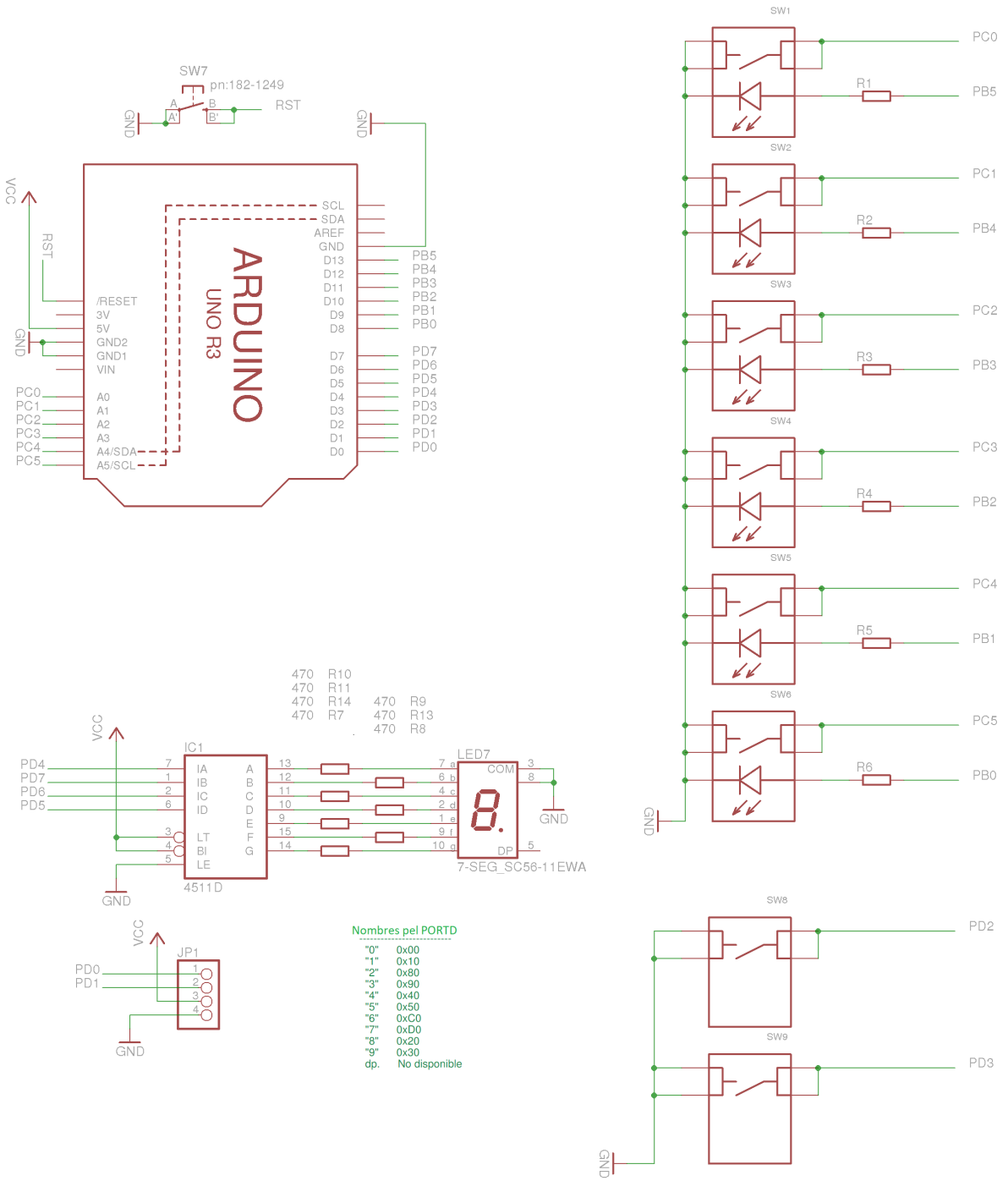


Figura 2: Esquema del *shield* BCAB.

**rep 'Ax'** Si rep els caràcters `'Ax'` on `x` pot ser 0,1,2,3,4 o 5, apaga el led corresponent (si estava encès).

**rep 'Dx'** Si rep els caràcters `'Dx'` on `x` pot ser 0,1,2,3,4 o 5, visualitza aquest valor al display de 7 segments.

**envia 'Bx'** Si es polsa el botó 0,1,2,3,4 o 5 s'envia `'Bx'` on `x` pot ser 0,1,2,3,4 o 5.

**envia 'OP'** Si es polsa el botó obrir porta s'envia `'OP'`.

**envia 'TP'** Si es polsa el botó tancar porta s'envia `'TP'`. En tots el polsadors, cal tractar convenientment el bouncing.

Una vegada dissenyay i construit aquest programari, proveu-lo usant un emulador de terminal sobre el port sèrie com és habitual.

### 3.2 Estudieu més Erlang

Hi ha diferents formes per poder connectar un dispositiu sèrie a Erlang:

- Opció A: Amb les comandes d'Erlang.

  open_port("Nom del port, **[**Opcions**]**), que retorna l'identificador del port que s'ha obert.

  port_connect(P, Pid), que crea una connexió bidireccional al dispositiu del sistema operatiu que s'identifica amb el port sèrie on teniu l'Arduino connectat. Si es rep quelcom pel port P, envia un missatge al procés Pid amb el format:

  {P, {data, Missatge}}

  port_command(P, Dada), que envia la dada Dada al port P.

  port_close(P), que tanca el port P.

Estudieu les funcions `Erlang` erlang:open_port() i la resta de funcions sobre ports que trobareu al manual d'`Erlang`. Pareu especial atenció a port_connect().

Un exemple per obrir i transferir dades a un port sèrie, pot ser:

```
-module(proUSB).
-export([inici/0, ordre/2, loop/1, exit/1]).

inici() ->
        Port = open_port("/dev/ttyACM0", []),
        Pid = spawn(proUSB, loop, []),
        port_connect(Port, Pid),
        {Port, Pid}.

ordre(Port, Valor) -> port_command(Port,Valor).

loop(P) ->
        receive
```

```
                {P,{data, A}} ->
                        io:format("Ha rebut: ~p~n", [A]),
                        loop(P);
                exit -> io:format("Proces finalitzat~n",[]);
                Altre -> io:format("Fi ~p~n",[Altre])
end.

exit(Pid, Port) ->
        port_close(Port),
        Pid!exit.
```

Evidentment, falta el programa que s'ha de carregar a l'arduino per generar les transmissions de dades en el cas de prémer algun botó, i interpretar les dades rebudes per actuar conseqüentment.

```
%% Ordres per que funcioni
% {Port, Pid} = proUSB:inici().
% proUSB:ordre(Port, "1").
% proUSB:exit(Pid, Port).
```

Si premeu un botó de la botonera i reveu missatges d'aquest tipus: [0,0,0,0,..], segurament teniu la velocitat del port mal configurada. Recordeu que la velocitat, el nombre de bits, la paritat i el nombre de bits d'stop han de coincidir. El sistema operatiu deixa un valor per defecte al port. Per veure quin valor té podeu utilitzar la comanda: stty -F /dev/ttyACMx i us dirà la velocitat (que és el més probable que no coincideixi). Si la voleu modificar, podeu utilitzar: stty -F /dev/ttyACMx 9600 per deixar-la a 9600 bits/s

- Opció B: Hi ha hagut gent que ha implementat llibreries per utilitzar el port sèrie. Totes són molt interessants i comporten un valor afegit. Us dono algunes referències:

1. erlang-serial
2. gen-serial
3. Amb Python
4. Un altre amb python

### 3.3 Conduïu la botonera des d'un procés

Creeu un proces `Erlang` senzill que obri una connexió al port sèrie i a través d'aquesta esperi que es polsi qualsevol polsador i, posteriorment vagi encenent cada led en un cicle de 2 segons cada un.

### 3.4 Implementeu un nou bcab

Implementeu ara un nou mòdul `bcab` amb exactament la mateixa API que el mòdul que havieu implementat. En aquest nou mòdul, però, la interfície d'usuari no es fa a través d'una finestra a l'escriptori sinó a través de la botonera de l'Arduino. Naturalment el nombre de pisos està limitat al nombre de polsadors/leds, és a dir 6.

Una vegada tingueu el nou mòdul `bcab`, empelteu-lo en el codi de la pràctica 5 i aconseguiu que funcioni correctament tot el sistema de l'ascensor.

### 3.5 Ascensor robust

Modifiqueu els mòduls de l'ascensor per tal d'aconseguir que sigui un sistema totalment robust sobre finalitzacions inesperades d'algun dels seus mòduls.

Al disposar de mòduls de diferents s'ha d'actuar de diferent forma:

- Si el mòdul representa una simulació d'una part de l'ascensor: motor, sensor, ContBP i bpis, només s'ha de controlar la seva finalització inesperada. Per exemple que algú tanqui la finestra o un dels processos es mori. El que s'ha de fer en aquests casos, és tornar a iniciar el procés que hagi tingut algun problema i fer un reset del sistema. Podeu simular la mort del motor, el sensor o el ContBP enviant un missatge que els finalitzi. El tancament d'una botonera de pis, el podeu simular tancant la botonera. Penseu a deixar alguna forma per finalitzar tot el procés ascensor, per exemple, enviant a l'ascensor un missatge de tancament del sistema.

- Si el mòdul és una part real del sistema, com la botonera de l'ascensor i per exemple, es desconnecta l'Arduino del port USB, representa que hi ha hagut un problema a la cabina i s'ha d'avisar de l'avaria i deixar el sistema aturat fins que s'arregli (es torni a connectar). La forma de donar l'avís, pot ser visualitzant un missatge a les botoneres de cada pis, dient que no funciona correctament. Un cop s'ha resolt el problema, s'ha tornat a connectar l'Arduino al USB, el sistema retorna a la normalitat.