

Informe pràctica 2

Eduard Martín Graells
Joan Peracaula Prats

March 2018

1 Introducció

En aquesta pràctica hem tingut un primer contacte amb els ports d'Entrada/Sortida de propòsit general (anomenats GPIOs) del microcontrolador. A més, hem refrescat conceptes de programació del flux d'un programa com són els bucles i els salts condicionals, en el llenguatge C.

En particular, hem hagut de modificar el comportament de diferents LEDs en funció del *Joystick* y dels botons *S1* i *S2*. En un primer moment hem hagut de canviar el color que emeten els LEDs RGB situats a la placa superior i més endavant fer diferents progressions amb els LEDs de la placa inferior.

2 Recursos utilitzats

Per realitzar aquesta pràctica hem utilitzat els components d'Entrada/Sortida de la placa *Boosterpack MK II*. En particular hem utilitzat el *Joystick*, tenint en compte totes les accions possibles, és a dir, els pins P4.5 - P5.7 (dreta-esquerra), P5.4 - P5.5 (amunt-avall) i el centre (prémer) amb pin P4.1; la pantalla LCD per tal de controlar les diferents accions i imprimir en quin mode estem; els polsadors S1 i S2, en els pins P5.1 i P3.5 respectivament, i per últim els LEDs, d'una banda els RGB: P2.6 (vermell), P2.4 (verd), P5.6 (blau) situats a la placa *Boosterpack MK II*; i per l'altre banda els 8 LEDs de l'*Adaptador MSP432-Bioloid* connectats al port P7, lògicament, un en cada un dels pins del port.

3 Configuració dels recursos

La configuració i inicialització dels botons, el *Joystick* i els LEDs RGB ja estava feta, tot i així, els comentarem.

Per als botons *S1* i *S2* el que hem fet primerament es elegir la funcionalitat que volem que tinguin amb el registre PxSELi (on x és el port i 'i' el bit 1 o 0), en aquest cas I/O. Com volem que siguin entrades (és el medi exterior el que els activa), els configurem com a tals amb PxDIR. Com que volem que quan s'activin generin una interrupció, així ho indiquem, amb el PxIES (indicant que la interrupció es de *Low to High* voltatge i després activem la interrupció amb PxIE. Per últim, netegem el flag de les interrupcions *setejant* la posició del vector corresponent del pin a 0.

Pel *Joystick*, és el mateix procediment que pels botons ja que és una entrada amb interrupció per cada moviment. Simplement cal prestar atenció a configurar els pins de cada moviment del component.

Pels LEDs, és diferent ja que són components de sortida. Pels RGB (i també pels vuit LEDs del port 7), només hem hagut de indicar que són sortides (recordem que aquests són GIPOs estrictament) amb `PxDIR` i *setejar-los* a 0 amb `PxOUT` per inicialitzar-los a apagats.

Per controlar la pantalla LCD s'utilitzen funcions ja donades a la llibreria *lib_PAE2*.

4 Funcions dels recursos

El que hem fet és que al utilitzar un dels botons o el *Joystick*, el sistema entri en una rutina específica per la interrupció del port, on la tractarem identificant el component que ha generat la interrupció i actualitzant la variable *estado* segons qui hagi interromput el procés. La variable *estado* ens serveix per escollir i tractar l'acció a realitzar després d'una interrupció.

Aleshores, un cop s'ha actualitzat el valor de *estado*, aquest deixa de ser el mateix que *estado_anterior* i això ens permet entrar a l'*if* on actualitzem la pantalla LCD i a més, mitjançant un *switch*, escollim la següent acció a realitzar pel robot en funció de *estado*. Aquestes accions tenen a veure en realitzar canvis als LEDs, ja siguin els RGB o els del port P7. Com que el tracte dels RGB és bastant senzill hem decidit fer-ho dins del *case*, en canvi pel tractament dels LEDs del port 7 hem preferit realitzar crides a funcions auxiliars ja que el seu tractament necessita de recorreguts (*for*) i condicionals (*if*) i hem cregut adient modular-ho.

5 Problemes

Degut a que aquesta pràctica és com una continuació de la primera pràctica introductòria, no hem tingut problemes gaire greus. En un principi, al seguir treballant amb un fitxer antic, hem tingut problemes amb el *debug* perquè després d'una interrupció no es cridava a les funcions específiques de tractament d'aquestes. En un primer moment vam creure que la configuració dels components no era correcta o que alguna cosa malament havíem fet, però més endavant ens vam donar compte que les capçaleres dels mètodes no estaven ben escrites.

També hem tingut una certa dificultat amb l'ús del sistema hexadecimal per referir-nos als pins que ens interessaven d'un port. Creiem que és degut al poc ús pràctic d'aquest sistema, però que confiem que anirem millorant.

D'altra banda, ens hem donat compte per un error, que si no s'inicialitza un pin, no es pot utilitzar, al oblidar-nos de cridar la funció *config_P7_LEDS*.

6 Diagrama de Flux

A continuació afegim una imatge del diagrama de flux d'aquesta pràctica. A més, també l'adjuntem com a fitxer extern a l'informe.

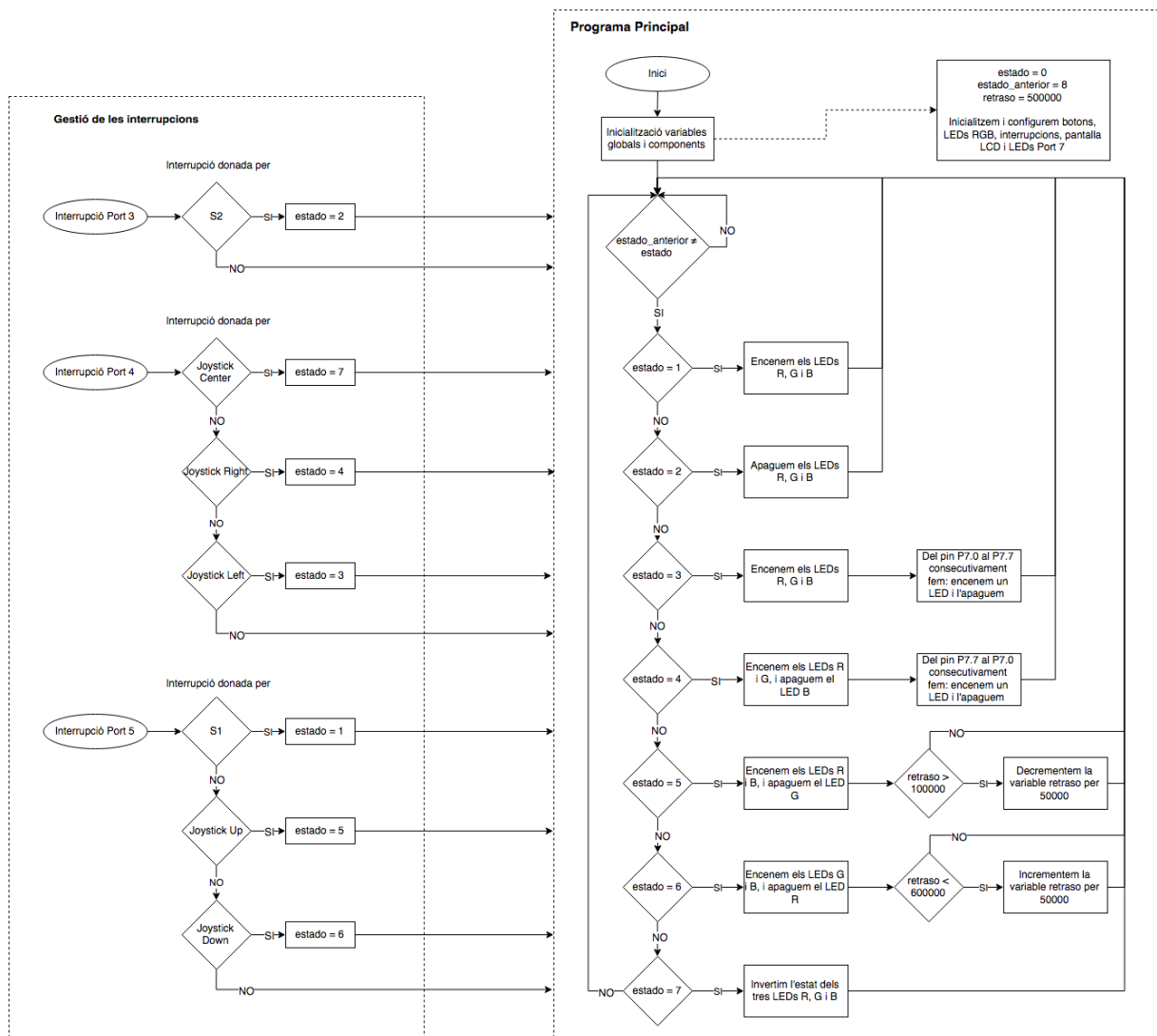


Figure 1: Diagrama de Flux

7 Conclusió

En aquesta pràctica hem estudiat i tractat amb components de tipus GPIOs. Això ens ha servit per aprendre a programar la seva configuració, inicialització i realitzar canvis del seu comportament. Els GPIOs d'entrada ens han permès treballar amb les interrupcions i ens ha ajudat a entendre el funcionament d'aquestes.

En particular hem après que quan es dona una interrupció es para el flux del programa principal per tractar-la. I per tal que aquest esdeveniment aïllat afecti al programa principal el que fem és actualitzar el valor d'una variable global que analitzem en el bucle infinit del *main*. Un altre aspecte important és que quan es dona una interrupció cal desactivar la resta d'interrupcions d'aquell port perquè quan estiguem tractant-la no se'n doni cap altra. Al final del tractament cal reactivar les interrupcions d'aquell port.

Pel que fa la metodologia, hem experimentat la importància d'utilitzar *breakpoints* i visualitzar els valors de variables i expressions durant el *debug*. Aquestes eines ens han permès agilitzar el procés de detecció i correcció d'errors.