

Configuració de Ports - Informe

Explicació de la pràctica 2

L'objectiu principal en aquesta pràctica serà principalment anar introduint com es controlen els ports d'Entrada/Sortida de propòsit general, o GPIOs (General Purpose Digital Input/Outputs) del microcontrolador que fem servir al laboratori. A més aprofitarem per anar introduint-nos al llenguatge C de programació a baix nivell, i a l'ús de diverses instruccions que ens permeten controlar el flux del programa, per fer salts condicionals i bucles mitjançant estructures if, for, do...while i switch/case.

Aquesta pràctica ens ha servit sobretot, per a familiaritzar-nos amb els conceptes bàsics de la programació del microcontrolador, i per aixó hem fet servir els recursos més "visuals" per a veure el funcionament de la placa. Hem utilitzat els diferents botons (direccions del joystick i botons S1/S2) per als 7 diferents estats (veure la Figura 1), i els leds, tant de la placa del microcontrolador (els leds RGB) com els de la placa secundària acoblada (8 leds que emeten llum vermella). També hem fet servir la pantalla per a veure en quin estat ens trobàvem.

Estat	LED B	LED G	LED R	Botó
1	1	1	1	S1
2	0	0	0	S2
3	1	1	1	Left
4	0	1	1	Right
5	1	0	1	Up
6	1	1	0	Dn
7	Invertir	Invertir	Invertir	Center

Figura 1. Taula de recursos

Pel que fa a com es configuren els diferents recursos, en el nostre codi de la pràctica ja venien implementades la majoria de les diferents funcions necessàries per a fer servir el nostre programa i poder carregar-lo a la memòria flash (reprogramable) del nostre robot, tot i que d'altres les havíem d'implementar nosaltres per tal de poder entendre globalment (tot i que encara no amb molta profunditat) el que fa el nostre executable. Les funcions per inicialitzar els recursos, com les interrupcions (`init_interrupciones()`), i tot el referent a la pantalla (`init_LCD(void)`, `borrar(uint8_t Linea)`, `escribir(char String[], uint8_t Linea)`) ja estaven escrites i no vam haver de tocar-les, nosaltres però, havíem d'implementar les funcions; `delay_t(uint32_t temps)` equivalent a la comanda "sleep" de línia de comandes, per tal de fer esperar un temps determinat abans d'executar una altra instrucció, la funció `config_P7_LEDS (void)` per preparar els pins del Port 7 com a sortides i configurar-los per a que inicialment estiguin apagats, de la mateixa manera tots els pins GPIO necessaris estaven configurats per tal de fer-los servir en altres funcions (LED RGB, botonera del

joystick i botons S1/S2, en total en aquesta pràctica hem fet servir pins dels ports 2, 3, 4, 5, 6 i 7), `PORTX_IRQHandler(void)`, per a manejar les interrupcions que donava cada botó i restablir-les al final del mètode, i jo he fet un parell de funcions extra (les quals no estan declarades a la llibreria del Campus Virtual), una per apagar els leds just abans de realitzar una acció anomenada `off_RGB()` i `knightRiderEffect()` per a emular l'efecte d'un recorregut d'esquerra a dreta / de dreta a esquerra dels leds vermells de la placa secundària en cas de el joystick es desplaçés cap a la dreta o cap a l'esquerra respectivament. A més havíem de preparar la funció main mitjançant una estructura switch per tal de, a partir de cada estat activar uns leds i cridar una funció o un altre, segons la taula d'estats que hem vist amb anterioritat (els diferents estats estan definits com a constants al principi del codi, on s'inicialitzen les diferents llibreries i variables globals necessàries). He deixat comentada la part que fa que el led RGB faci llumetes, ja que personalment em molestava per a veure bé el que estava fent el led. Val a dir que totes les funcionalitats han estat implementades amb èxit donat a que s'ha revisat la documentació proporcionada al Campus Virtual, com el manual de la placa de Texas Instruments.

Alguns dels problemes que m'han sorgit, i que s'han arreglat fent un debugging pas per pas del codi que he anat desenvolupant de forma autònoma, han sigut per no llegir del tot l'enunciat de la pràctica (ja que en aquesta pràctica s'especifica pas per pas tot el que s'ha de fer), com el fet de configurar el Port 7 per tal d'inicialitzar els pins de la placa secundària per a que funcionessin de manera correcta, i l'implementació de les funcions `PORTX_IRQHandler(void)` ja que al principi no tenia del tot clar com funcionaven. En general no hi ha hagut molta problemàtica tot i aquestes qüestions anteriorment citades.

Per a resumir, aquesta pràctica ha anat molt bé al conjunt de l'alumnat per entendre els principis bàsics que fa servir una 'Arquitectura encastada' com és el nostre microcontrolador. S'explica de forma progressiva, com utilitzar els nostres recursos (pantalla, botonera del joystick, botons S1/S2, conjunt de LEDS), el funcionament mitjançant ports i pins de la nostra placa, com s'utilitzen els diferents registres per a dur a terme les funcions i com funcionen els pins GPIO en concret.

El codi de la pràctica està degudament comentat i es troba adjunt en aquesta carpeta.

A continuació més avall mostrarem el diagrama de flux del programa d'aquesta segona pràctica.

Diagrama de flux

Em centraré en la funció main del codi de la Pràctica on trobem el switch (Figura 1. Taula d'estats) per a resumir el flux del programa.

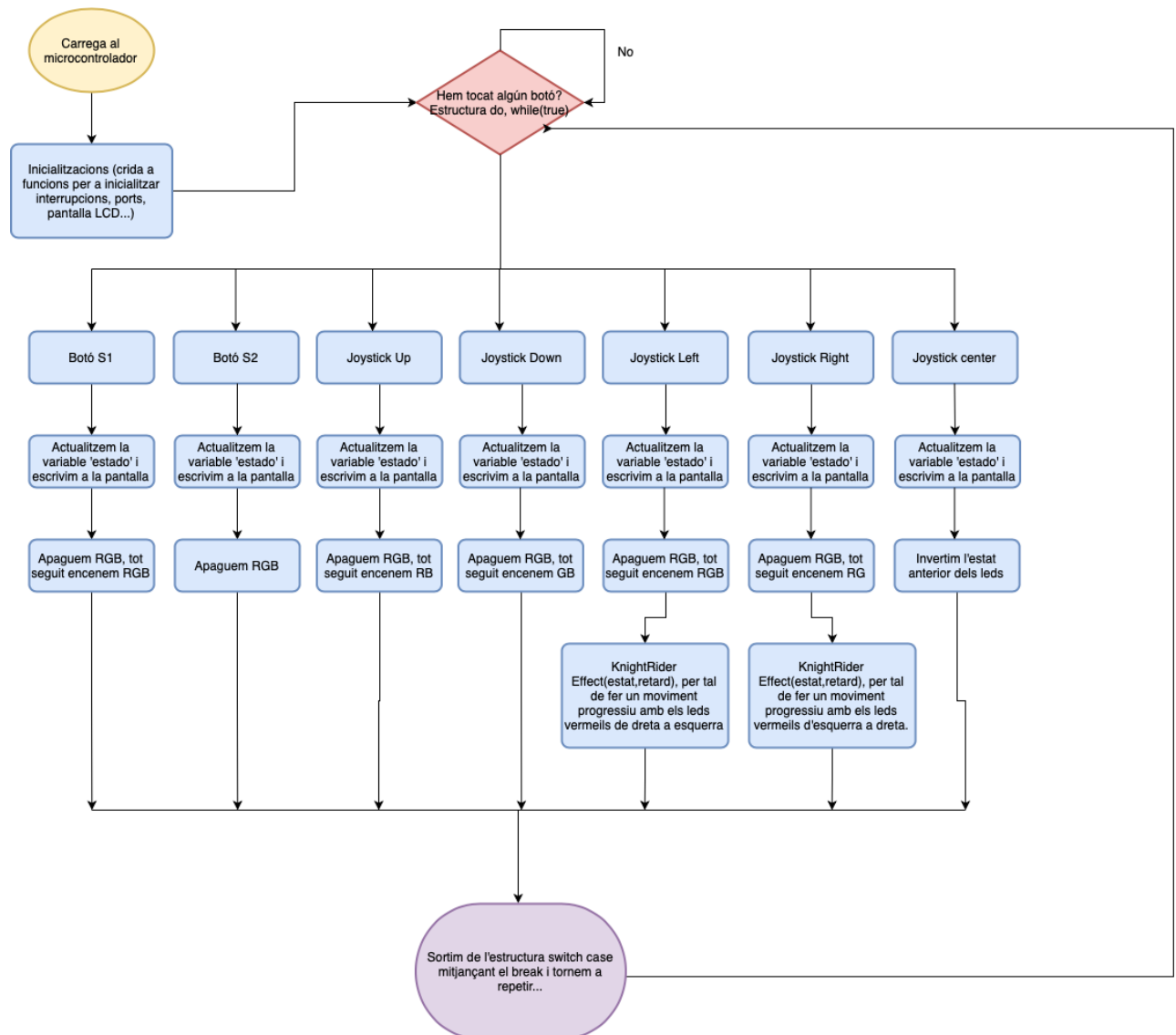


Figura 2. Diagrama de flux del programa de la segona pràctica