

Pràctica 6

Programa de control del robot

La nostra proposta de pràctica de moviment del robot cobreix tots els casos enunciats a la pràctica així com l'apartat opcional del LCD. Tot seguit llistarem els diferents blocs en que es pot dividir el codi. Al final de la memòria hi ha el llistat complet. Cal dir que tot el codi adjuntat funciona perfectament i ha estat comprovat al robot del laboratori.

Hem afegit un arxiu anomenat *lcd.c* per al control del LCD així com un *mis_includes.h* on es trobem macros auxiliars així com totes les constants que el programa fa servir. D'aquesta manera només caldrà canviar les definicions i no la resta de codi.

Recepció de les dades

La recepció de les dades es fa amb el port sèrie RxD0 i la seva configuració inicial es troba a la rutina **InicialitzaSerie**. Bàsicament el que fem és definir el vector d'interrupcions, inhabilitar els errors diversos que ens puguin generar interrupció i definir el mode de funcionament.

El mode triat és 8 bits de dades, sense bit de paritat i amb un bit de stop. Si nosaltres enviem les dades de la forma [A,B,C,D,1,1,1,1] el registre de dades rebudes del canal 0 (RxB0) prendrà el valor de [1,1,1,1,D,C,B,A]. És per aquest motiu que els codis del comandament estan definits girats al fitxer *mis_includes.h*.

Respecte al tractament d'errors nosaltres hem decidit no utilitzar-lo ja que trobem que no aporta res el fet de saber que s'ha rebut un codi invàlid. Podria succeir (com vam experimentar) que donat que hi ha un obstacle o bé movem el comandament es rep un codi que és vàlid (que té una ordre associada) però que no és el de la tecla que estem pitjant. Aquests errors no es poden arreglar de forma trivial. Una possible solució seria modificar el comandament per a que enviés una seqüència com per exemple [A,B,C,D,A,B,C,D] i la probabilitat que variïn els bits formant una seqüència vàlida és menor. Per altra banda sí que comprovem i exigim a les dades rebudes que tinguin els últims 4 bits de stop a 1. Si això no es compleix rebutgem la dada per invàlida.

Tractament de les comandes

El tractament de comandes es fa de mode immediat al rebre-la i s'encarrega al timer 0 la feina d'aturar aquest moviment en cas de no rebre cas més dada. En el cas de guardar o esborrar el tractament és guardar la posició en un vector, en cas d'esborrar només cal invalidar les dades del vector. El cas de la reproducció és bastant diferent. Caldrà comptar les passos, però haurem d'ignorar les dades rebudes pel comandament així com evitar que el timer0 aturi el robot passats 20ms a la vegada que modulem la velocitat dels motors amb el timer.

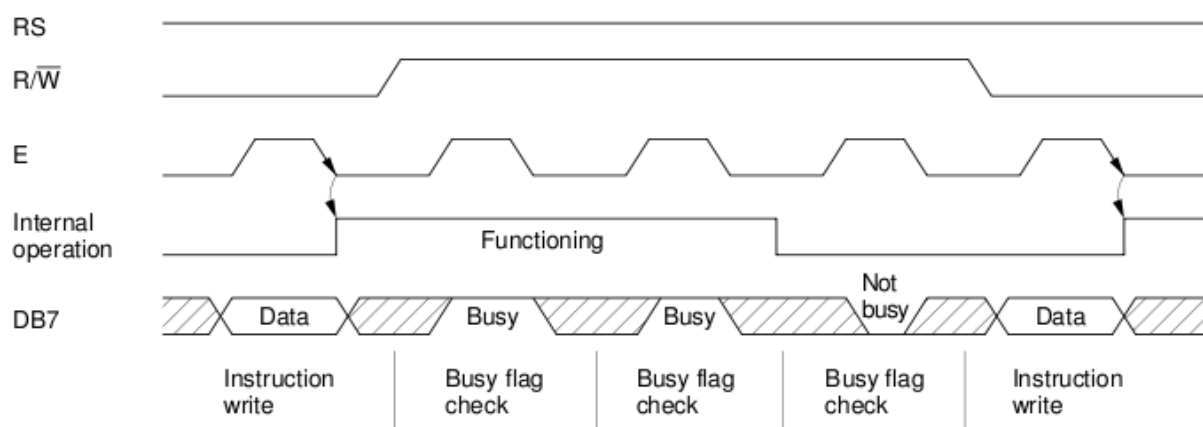
Mode de reproducció

La solució més senzilla i ràpida és tenir una variable que ens ho indiqui i utilitzar uns ifs per a canviar el comportament dels timers en el cas de la reproducció. A més farem que el *main* sigui l'encarregat de calcular quin moviment cal realitzar per a portar el robot a la posició desitjada. Per a fer això només cal comprovar quina és la posició actual i quina és la següent, en funció de si és major o menor haurem de moure'ns cap endavant o enrere. Anirem comprovant això contínuament fins que arribem a la posició. Unes variables globals indicaran com ens hem de moure i el timer serà l'encarregat de moure els motors. En cas que ens apropem a la posició desitjada caldrà frenar el motor i per això definirem unes variables de cycle de treball que ens permetran fer funcionar els motors més lentament. El timer doncs, activarà cada un dels motors segons els seu cycle de treball que serà 100% normalment i el 25% en apropar-nos a menys de 5 passos del final. La modulació es realitza aturant el motor el 75% del temps i activant-lo el 25% restant. És important notar que a l'enunciat diu que el període sigui de 80ms, però la nostra experiència ens diu que un període molt menor com 40 o 20ms.

LCD

Com a part opcional de la pràctica hem fet servir la pantalla LCD disponible al port 0 del microcontrolador. El codi el podem trobar a *lcd.c* comentat. Bàsicament el que fem és inicialitzar el LCD amb una seqüència que trobem explicada al manual d'HITACHI i llavors definim unes funcions que escriuen frases a la pantalla. Aquestes funcions situen el cursor a l'origen i escriuen caràcter a caràcter la frase.

Per a que tot això funcioni després d'enviar una dada cal garantir un temps a la senyal enable així com un temps d'execució d'instrucció. Això es veu al cronograma corresponent al manual. Notem que fem servir el mode 8 bits, amb el que les dades s'envien en un sol cycle.



Com podem veure cal enviar la dada i al cap de 40µs aproximadament hem de desactivar la senyal enable i la senyal de write i esperar el temps d'execució d'una instrucció (uns 40µs però depèn de la instrucció).

A continuació el llistat de codi complet. Els fitxers que no es defineixen són els mateixos que ens donàveu com a enunciat.