

CONNEXIÓ AL ROBOT**Creació de les Funcions Bàsiques de Comunicació i d'una Llibreria de Funcions per Controlar el Robot.
(3 sessions)**

Aquesta pràctica la dividirem en dos parts:

- Generar els recursos bàsics per comunicar la placa del microcontrolador amb el Robot.
 - Fer una funció per configurar les comunicacions amb els mòduls del Robot.
 - Fer una funció per Enviar “comandaments” als mòduls del Robot.
 - Fer una funció per Rebre informació dels mòduls del Robot.
- Fer una llibreria de funcions que permetin controlar el robot.

1.- Recursos bàsics per comunicar-se amb el robot.

Com ja sabem, la comunicació entre el microcontrolador i els mòduls del robot (AX-12: Motors, AX-S1: Sensors) es fa mitjançant un protocol sèrie asíncron. El recurs del processador que s'encarrega d'això es la USCI (*Universal Serial Communication Interface*) programada com a UART (*Universal Asynchronous Receiver Transmitter*). El nostre microcontrolador disposa de 4 USCIs que poden funcionar com a UART, de les quals nosaltres farem servir la UCA2, que està al port 3 (pins P3.2 = RX, P3.3 = TX).

Necessitareu diverses funcions que heu d'introduir al vostre programa, que són les bàsiques per fer la comunicació:

- A. Configuració del sistema de rellotge del microcontrolador UCS (*Unified Clock System*) per que la UART treballi a la freqüència requerida pels mòduls Dynamixel.
- B. Configuració de la UART.
- C. Funcions de transmissió i recepció d'una trama tal com la requereixen els mòduls Dynamixel.

1.A- Configuració de la UCS

La UCS configura 3 fonts de rellotge:

- **MCLK:** *Master Clock* per a la CPU.
- **SMCLK:** *Secondary Master Clock* disponible pels perifèrics.
- **ACLK:** *Auxiliary Clock* disponible pels perifèrics.

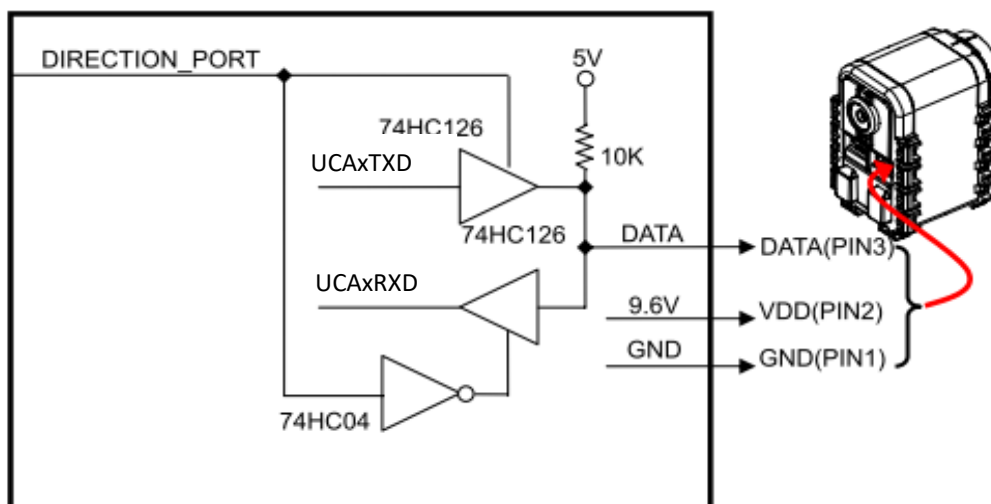
Com ja ho hem estat fent en les pràctiques anteriors, utilitzarem la funció `“init_ucs_24MHz()”`, disponible a la llibreria `“lib_PAE2.h”`.

Amb aquesta funció, l'ACLK ens queda a 2¹⁵Hz, i l'SMCLK funciona a 24MHz.

Penseu que la configuració de la UCS repercuteix en el funcionament de tot el sistema, per tant es recomana que aquesta configuració es faci al començament de l'execució del programa (just després d'aturar el “Watchdog Timer” tal com hem anat fent en les pràctiques anteriors).

1.B- Configuració de la UART

- La configuració de la UART ha d'ajustar-se al sistema de comunicació dels mòduls Dynamixel. El protocol que fan servir aquests mòduls és sèrie, asíncron de 8 bits, 1 bit de stop i sense paritat. Les comunicacions dels mòduls Dynamixel de la pràctica funcionen amb un *Baud rate* de 500kb/s. No obstant, per millorar la detecció dels bits en recepció, farem treballar la nostra UART amb sobremostreig (oversampling), agafant 16 mostres per cada bit transmès. Això vol dir que una UART que vulgui treballar a aquest *Baud rate* ha d'estar alimentada amb un rellotge 16 vegades més ràpid. Tenint en compte tots aquest requeriments, estudeu la configuració de la UART a partir dels valors assignats als registres de configuració adients en la funció "`init_uart()`" proporcionada a classe de teoria, i expliqueu-lo a l'informe (recordeu que com a part de les eines de depuració, el CCS disposa d'una pestanya "Registers", que ens permet inspeccionar el continguts dels registres del microcontrolador).
- Un altre punt a tenir en compte és que els mòduls Dynamixel fan servir comunicació "Half-duplex", és a dir que només tenen una línia "Data" que serveix tant per transmetre com per rebre dades, mentre que la UART del microcontrolador és Full-duplex: té una línia per transmetre UCxTXD i una altra per rebre UCxRXD. Per commutar la connexió d'aquesta línia "Data" a TX o a RX, a la placa d'interfície inferior (**Adaptador MSP432-Bioloid**) tenim un circuit que passa de dos línies a una i viceversa:



Hem de tenir en compte que ara des del microcontrolador, per programa, hem de controlar el senyal "**DIRECTION_PORT**". A la nostra placa l'hem connectat al pin GPIO **P3.0**. Hem d'inicialitzar aquest senyal i gestionar-ho en funció de si volem enviar o rebre missatges a les funcions corresponents de transmetre o rebre.

Per això farem un parell de funcions que controlin aquest pin i les farem servir just quan comencem una transmissió o una recepció.

- També és evident que per una línia Half-duplex, no hi poden circular dades de transmissió i dades de recepció al mateix temps. A més a més, cada cop que enviem un comandament a algun mòdul (TXD des del punt de vista del microcontrolador), aquest sempre torna una resposta (RXD des del punt de vista del microcontrolador), com a mínim amb un missatge informant de que ha rebut les dades sense error. És un punt important que haurem de tenir en compte als nostres programes, per evitar "col·lisions" a la línia "Data".

Finalment, aquesta línia "Data", l'haurem de mantenir sempre per defecte en l'estat RXD. La canviarem a TXD només quan vulguem transmetre-li dades a algun mòdul, i la tornarem a RXD quan s'hagi acabat de transmetre l'últim bit de la trama a enviar.

1.C- Funció de transmissió d'una trama tal com la requereixen els mòduls Dynamixel.

El microcontrolador es comunica amb els mòduls Dynamixel enviant i rebent paquets de dades, o trames. Hi ha dos tipus de paquets: "paquets d'instruccions" (enviat des del microcontrolador als mòduls Dynamixel, TXD) i "paquets d'estat" (enviat des dels mòduls Dynamixel al microcontrolador, RXD.)

Les funcions proporcionades a classe de teoria serveixen per transmetre una trama d'instruccions amb el format dels mòduls Dynamixel.

- A partir del que s'ha estudiat a la classe de teoria, i dels *Manuals* del AX-12 o el AX-S1, analitzeu les funcions de transmissió de dades (TXD) i estudeu el seu funcionament. **Comenteu les instruccions que no ho estan i expliqueu tot això a l'informe de la pràctica.**
- Feu una funció similar per tal de rebre dades (RXD) dels mòduls Dynamixel, utilitzant en aquest cas la capacitat de la USCI de generar interrupcions en recepció.

2.- Fer una Llibreria de Funcions per Controlar el Robot.

Es tracta de fer un conjunt de funcions per tal de controlar el robot. Per exemple, moure cap endavant, moure cap enrere... canviar velocitat, moure contínuament, moure uns determinats graus, capturar els diferents tipus de sensors...

Per fer aquesta llibreria feu servir les funcions de l'apartat anterior i la documentació dels mòduls Dynamixel AX-12 i AX-S1 que teniu al camp virtual.

Per fer aquesta pràctica, estudeu amb detall el que s'ha explicat a la classe de teoria i que teniu al campus virtual. I, evidentment, el capítol corresponent a la *USCI* que feu servir al *Technical Reference Manual* del nostre microcontrolador, ja que haureu de configurar els registres adients.