Enunciat de la pràctica.

1) Traduïu a (o bé "codifiqueu en") "C" l'apartat **'1'** de la pràctica 5 (feta en assemblador). Això és, un 'main' que consisteix en un bucle infinit que crida indefinidament a 'PutChar('A')'. Feu servir el model de memòria 'tiny', no calen variables i compteu amb el port sèrie inicialitzat pel carregador. D'ara endavant, quan tingueu dubtes (p. Ex. La placa falla?) confirmeu-ho amb aquest programa.

Feu un programa en el que es vegi l'ús de la subrutina 'EscriuCadena()' basada en l'existència de 'PutChar()' per tal d'utilitzar la línia sèrie com a "canal de depuració" en successives etapes. 'EscriuCadena("Hola mon")' funcionarà perfectament si teniu en compte que la cadena és del tipus 'char *cadena' i acabada (el compilador se n'encarrega) amb un caràcter 'NUL' (un "número zero"). Podeu accedir als elements de la cadena amb un índex ('char i' vuit bits) de la forma 'carac=cadena[i];'. Aquest codi es pot fer servir al inici dels successius exercicis per verificar que la línia sèrie va bé.

2) Passeu a "C" els apartats **'2'** i **'3'** de la pràctica 5 (feta en assemblador). Potser resulta més senzill reescriure'ls de nou (partint de zero) que "traduir literalment" les sentències assemblador. Les 'macros' del assemblador es construeixen de la següent forma en "C":

Definició "ASM":	**Definició "C"**
MACRO %pols_led	#define pols_led(param) {param=1;Retard();param=0;}
SETB \0	
CALL Retard	
CLR \0	
ENDMAC	
Instància "ASM"	**Instància "C"**
%pols_led P3.5	pols_led(P3.5);

3) Feu un programa que faci servir la línia sèrie per interrupció. Cal que inicialitzeu vosaltres la línia sèrie. Implementeu la rutina de servei a la interrupció per rebre caràcters i emmagatzemar-los en una cua circular (la estructura de cua està feta i disponible a la Web). La transmissió serà "directa" i amb "espera activa" (no s'emet un caràcter fins que l'anterior no "ha sortit"). D'aquesta manera, el programa principal (o altres rutines) anirà a recollir els caràcters a la cua sense preocupar-se de la gestió del port. Amb això il·lustrem un problema de productors (la RSI) / consumidors (el programa principal) d'informació. Aquest programa ha d'emetre les "paraules" (seqüències de caràcters separades per espais) a mida que van arribant.

Podeu fer servir l'estructura de 'cua' d'en Jordi Ferrer que teniu a la web. Tal i com indica el seu autor, sentiu-vos lliures de afegir i modificar qualsevol cosa que creieu convenient. En aquesta 'versió' s'han eliminat **intencionadament** certes porcions de codi per evitar que caigueu en la temptació de fer 'cut i paste' sense entendre el sentit del que esteu fent.

4) Trames punt a punt del tipus: 'Inici- Comanda- Valor/...s- Final'. Feu una estructura de dades (tipus i operacions) que permeti gestionar trames de caràcters de format i mida fixa. Aquesta estructura ha de tenir:

- **TIPUS Trama.** Caràcter d'inici (per exemple '$'), Caràcter de comanda, Caràcter/s (1 ó 2) de Valor, Caràcter de fi (p.e. RETORN). Això és una "descripció funcional" i NO representa cap tipus d'emmagatzemament.

- **FUNCIO EnquestaTrama() RETORNA booleà.** Funció que retorna "cert" si s'ha rebut una Trama vàlida (segons especificació anterior una trama és vàlida si s'ha rebut 2 ó 3 caràcters entre un "inici" i un "final"). Aquesta funció s'anirà cridant periòdicament des del programa principal i consultarà la "cua" de recepció; amb això s'aconsegueix la "enquesta" (o poolling) a nivell de trama. Ajudeu- vos d'una variable estàtica ('static char estat_trama' que no perd el valor en crides successives a la funció) i assigneu- li el "sentit" 0 "esperant inici", 1 "esperant comanda", .. n "esperant final". Quan es reb un caràcter s'incrementa aquesta variable en una unitat si es compleixen les condicions necessàries. D'aquesta manera construïm una 'màquina d'estats', reflex de l'estat en que es troba la recepció d'una trama formada per caràcters (que van arribant al llarg del temps assíncronament). Teniu en compte les següents regles: Qualsevol caràcter "*inici*" marca un inici de trama (i, per tant, posa a '1' la màquina d'estats); el primer caràcter diferent de "*inici*", un cop iniciada la trama, és "*comanda*"; la trama s'acaba i és vàlida si es rep el caràcter de "*final*" al seu lloc; NO apareixeran caràcters "*inici*" ni "*final*" dins de "*comanda*" ni "*valor*" (en successives pràctiques, si cal, modificarem aquest codi per contemplar estratègies 'DLE').

- **FUNCIO ComandaTrama() RETORNA caràcter.** Funció que retorna el caràcter associat a la última trama rebuda.

- **FUNCIO ValorTrama() RETORNA caràcter.** Funció que retorna el valor associat a la última trama rebuda. Per evitar "problemes", aquesta funció esborra el 'flag" de 'NovaTrama' (o trama no llegida).

- **VARIABLE NovaTrama TIPUS bit.** Indica (quan és a "1") que ha arribat una trama sencera vàlida. Es posada a "cert" per 'EnquestaTrama()' i a zero (o fals) per 'ValorTrama()'.

- **VARIABLES Comanda, Valor TIPUS caràcter.** Escrites per 'EnquestaTrama()' emmagatzemen els caràcters rebuts (en el moment adient). Només són "consultades" (o llegides) per 'ComandaTrama()' i 'ValorTrama()'.

En aquest moment (més endavant ens serà molt més útil), la utilització d'aquesta estructura es pot fer servir (i així ho heu de fer) en un programa que governi l'encesa i apagada dels LEDs de la placa de pràctiques. Les comandes vàlides seran 'A' per apagar i 'E' per encendre. Els valors vàlids seran del '1' al '4' (caràcters ASCII, res de números).

Basant-nos en l'existència d'aquesta estructura, el programa descrit seria el següent (suposem que existeixen els '#defines' adients):

```
main()
{
        while(cert)             /* bucle infinit */
        {
                if ( EnquestaTrama() )      /* ha arribat una trama "sencera", la processem */
                                            /* aprofitem aquesta crida perque es vagi actualitzant */
                                            /* la màquina d'estats */
                {
                        switch ( ComandaTrama() ) /* Què cal fer? */
                        {
                                case 'A': apaga_led( ASCII2HEX( ValorTrama() ) );
                                break;
                                case 'E': encen_led( ASCII2HEX( ValorTrama() ) );
                                break;
                        }                   /* ASCII2HEX converteix els caràcters a números */
                }       /* cridar a ValorTrama() fa que EnquestaTrama() no retorni cert */
        }               /* fins que no s'ha rebut una trama nova (té sentit de "processada") */
}
```