

1- Obtener el código fuente del tp3 y verificarlos.

a. Leer los archivos fuentes del práctico. Entender el archivo cabecera serial.h y observar como es utilizado por main.c

b. Agregue un archivo Makefile al proyecto (puede utilizar el del tp2 modificando los nombres de los archivos del proyecto). Verifique el Makefile con

`make clean; make`

2- Desarrollar un driver (controlador) para el periférico UART del atmega328p, utilizando los archivos propuestos.

a. Comience completando la estructura de datos que hace "overlay" con los registros del hardware del USART del atmega328.

- Lea y comprenda utilizando la hoja de datos la ubicación de los registros (pagina 612 del manual del atmega328p).

- Estudie minimamente la descripción de cada registro del periférico (página 191 del manual del atmega328p).

b. Lea nuevamente el código fuente de main.c para comprender la manera en que main utiliza la API del driver.

c. Escriba la rutina de inicialización. Utilice un baud rate de 9600 bits por segundo, 8bits de datos, sin bit de paridad, y un bit de stop. Utilice el puntero a la estructura de registros para configurar el UART y activar también la recepción y transmisión.

d. Escriba las rutinas serial_put_char() y serial_get_char(), utilizando E/S programada.

3. Utilice cutecom o minicom (tal vez como root o con sudo, depende del sistema Linux) para comunicarse con el arduino pro mini. El dispositivo serial en Linux será del estilo /dev/ttyUSBX (puede utilizar el comando dmesg cuando conecta el arduino pro mini con el adaptador USB, para conocer el dispositivo correcto), y utilice los mismos parametros de comunicación que la aplicación embebida.

4. Ampliar el código fuente para que main espere un byte desde la PC:

- Si la PC envió la letra 'c' el sistema realiza el conteo con los leds de 0 a 7 (utilice el código del TP anterior).
- Si la PC envió la letra 'k' el sistema realiza el efecto Knight Rider un par de veces.