

## PSE. Trabajo práctico UART serial

1- Obtener el código fuente del tp serial y verificarlos.

- a. Leer los archivos fuentes del práctico. Entender el archivo cabecera serial.h y observar cómo es utilizado por main.c
- b. Agregue un archivo Makefile al proyecto (puede utilizar el de la web 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 atmega328p.
  - Lea y comprenda utilizando la hoja de datos la ubicación de los registros (pagina 612 del manual del atmega328p).
  - Estudie mínimamente 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` o `cat /dev/ttyUSB0` (tal vez como `root` o con `sudo`, depende del sistema Linux) para comunicarse con el arduino nano. 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 parámetros de comunicación que la aplicación embebida.

4. Ampliar el código fuente para que main espere pulsaciones de teclas desde la PC (bytes):

Si se presiona la tecla 'k' entonces se enciende/apaga el `knight_rider`.

Si se presiona la tecla 'u' el sistema embebido utiliza el ultrasonido y responde con la distancia del objeto al sistema.

5. Colocar dos pulsadores al sistema embebido. Convertir el sistema embebido en un gamepad.

Controlar un video juego en la PC Linux con los dos pulsadores. Si se presiona el pulsador de la izquierda el personaje del video juego se mueve a la izquierda. Si se presiona el pulsador derecho el personaje del video juego se mueve a la derecha. AYUDA: lo más sencillo es enviar bytes que representen ascii. Ejemplo, la 'a' para cuando pulsan el pulsador izquierdo, y el byte 'd' para cuando pulsan el pulsador derecho. De esta manera la PC obtendrá 'a' y 'd' según pulsación. Luego en la PC tendrán que tener algun programa de ustedes que al recibir esos bytes "inyecte" pulsaciones de teclas. Ver próxima página.

## PSE. Trabajo práctico UART serial

### NOTAS de AYUDA entre la interacción PC - MCU:

#### Ejercicio 4.

Pueden utilizar el lenguaje de programación que quieran para leer pulsaciones del teclado de la PC y enviar las mismas al atmega328p (java, c, python, rust, whatever). En particular, si nunca lo hicieron, deberían buscar cómo enviar bytes a un dispositivo “serial” en el lenguaje de programación que utilicen.

Ejemplo utilizando el lenguaje del shell (terminal en Linux):

```
stty -F /dev/ttyUSB0 speed 9600 # o 2400 la que necesiten
while true; do
    read -n 1 tecla                # espera por una pulsación del teclado, almacena el valor en 'tecla'
    echo $tecla                    # muestra la letra de la pulsación en pantalla
    echo $tecla >> /dev/ttyUSB0    # envía el contenido de tecla al driver del dispositivo serial,
                                   # quien a su vez lo envía por el hw al MCU del otro lado
done
```

#### Ejercicio 5.

Lo mismo que para el ej. 4. Pueden utilizar el lenguaje de programación que quieran. Aquí dejo una posibilidad usando el shell de Linux y el programa xdotool (paquete xdotool)

```
stty -F /dev/ttyUSB0 speed 9600 # o 2400 la que necesiten
cat /dev/ttyUSB0 | while read -n 1 tecla; do
    xdotool key $tecla
done
```

Ejecutan eso en una terminal. Hacen foco en la ventana que tenga el video juego. Presionan los pulsadores del sistema embebido para ver si surge efecto.

**Ejemplos sin corroborar. Están advertidos :)**