

## **1 Track#1 – ‘Software oriented’**

### **1.1 Get UART up and running.**

Use the BSP UART driver to make UART port 1 echo text. Look at the example `uart_polled` for inspiration (this example uses the BSP function `lpc2129_uart_polled` driver)

### **1.2 Use Graphics Library to create full screen effects and scrolling text.**

Look at the example `gfxlib_effect` for inspiration.

### **1.3 Use BSP and Graphics Library to set pixels/use effects according to UART data.**

Combine 1.1 and 1.2.

### **1.4 For a Prize: Use an RTOS to make X bouncing 4x4 balls that collides with each other (one task = one ball, use messages between tasks)**

Choose which RTOS to use (FreeRTOS or uc/OS-II). Investigate the examples in `rtos/FreeRTOS/Demo/ARM7_LPC2129_GCC` and `rtos/MICRIUM/SOFTWARE/EvalBoards/Philips/Bitfire/GCC/app` before making this choice. In general uc/OS-II is better documented on this CD.

## **2 Track #2 – ‘Hardware oriented’**

### **2.1 Setup a UART with 38.4kbps and echo characters.**

Do not use the BSP driver, but you can use it for inspiration! ☺ The BSP function is called `lpc2129_uart_polled`.

### **2.2 Add a timer interrupt and broadcast hello message on UART each sec.**

Read the datasheet to setup the Timer correctly. You may use the VIC driver in the BSP to setup the interrupts, use the example `uart_int` for inspiration.

### **2.3 Setup CAN peripheral and send a 11b message between the two channels**

Read the datasheet and look at the `can_polled` example for CAN tips.

### **2.4 For a Prize: Send a message from UART input trough a CAN network and scroll the sent message on the display.**

Build on 2.3 -- use CAN-0 and CAN-1 on same board for test.