

```

UASK_KEY = (xaddr) CH2_ASK_KEY_XADDR;          // defined in library.h

UEMIT = (xaddr) CH2_EMIT_XADDR;              // defined in library.h
printf("Ready to echo incoming characters on Channel2...\n");
while(1)
{
    this_char = _readTerminal();
    if( this_char == '\r')
        this_char = '\n'; // substitute linefeed for cr, ansi-c style
    putchar(this_char);    // automatically adds cr in front of linefeed
}

_Q void Run_Demo(void)
// builds and activates two forth-monitor tasks,
// each using a separate channel on the uart module.
{
    Set_UART_Number(UART_MODULE_NUM);
    if(Default_UART_Init(UART_MODULE_NUM)) // initialize the hardware
        printf("\nError: Invalid protocol combination was specified!\n");
    else
    {
        printf("\nStarting UART Module Demo...\n");
        SERIAL_ACCESS = RELEASE_ALWAYS; // ensure lots of PAUSES in Forth task
        NEXT_TASK = TASKBASE; // required! empty the round-robin task loop
        BUILD_C_TASK(0,0,&ch1_task); // no heap needed
        BUILD_C_TASK(0,0,&ch2_task); // no heap needed
        ACTIVATE(CH1_Monitor,&ch1_task);
        ACTIVATE(CH2_Monitor,&ch2_task);
        StartTimeslicer(); // enable task switching
    }
}

void main(void)
{
    Run_Demo();
}

```

Forth Demonstration Program

This section presents the ANSI C version of the demonstration program source code.

```

\ *****
\ FILE NAME:   UModDemo.4TH
\ copyright 2002 Mosaic Industries, Inc. All rights reserved.
\ -----
\ DATE:       5/14/2002
\ VERSION:    1.1, for QED4 or Panel-Touch Controller with WildCard Carrier Board
\ -----
\ This is the demonstration code for the Dual UART Module.
\ Please see the UART Module User Guide for more details.
\ The accompanying file named UModDvr.4th (or the corresponding kernel extension)
\ MUST be loaded before this file can be loaded.
\ This is an illustrative demonstration program that
\ shows how to initialize the uarts for RS232 operation and run dual
\ QED monitor tasks using the two UART Module serial ports.
\ -----
\ When the top level function Run_Demo is running, the QED Board
\ or Panel-Touch Controller is simultaneously using 3 serial ports:
\ the standard primary serial port and each of the two serial channels

```

```

\ on the UART Module is running an instance of the QED-Forth monitor.
\ Using the constants and/or the Default_UART_Init function
\ defined in this file, you may customize the
\ baud rate and protocol settings for the UART Module ports.
\
\ The QED operating system supports revectorable I/O, meaning that
\ in any given task the standard serial I/O routines such as
\ CR and ." can be made to use any specified serial channel.
\ All that is required is to customize and revector (store the xcfa of)
\ three functions named Key, ?Key, and Emit to the specified serial channel
\ for the specified task. This file shows how to do this
\ using the functions defined in the UART Module kernel extension.

\ MAKE SURE THAT THE UART_MODULE_NUM CONSTANT MATCHES YOUR HARDWARE JUMPER SETTINGS!!

\ -----
\
\ Demonstration functions defined in this file:
\ UART_MODULE_NUM \ MUST match hardware jumper settings!
\ Default_UART_Init ( module_num -- result ) \ demonstrates how to initialize module
\ Run_Demo ( -- )

\ -----
\ Notes:
\
\ Disclaimer: THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, WITHOUT
\ ANY WARRANTIES OR REPRESENTATIONS EXPRESS OR IMPLIED,
\ INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES
\ OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
\
\ *****

\ This version is for QED4 Boards, Panel-Touch Controllers, with WildCard Carrier Board.

HEX
WHICH.MAP 0=
IFTRUE 4 PAGE.TO.RAM \ if in standard.map...
    5 PAGE.TO.RAM
    6 PAGE.TO.RAM
    DOWNLOAD.MAP
ENDIFTRUE

\ set memory map on page 4 after load of UMod_Dvr.4th:
\ 800 4 DP X! 5800 4 NP X! \ comment this out or change it if memory map is already set up

F WIDTH ! \ set width of names stored in dictionary

ANEW UDemo_Code \ define forget marker for easy re-loading

\ ***** DEMONSTRATION PROGRAM *****

\ The default task runs FORTH as usual, using the 68HC11 UART.
\ We create a second task and a third task that also run FORTH,
\ each communicating using a serial channel on the UART Module.
\ To run this demonstration, simply execute:
\ RUN_DEMO
\ You'll be running FORTH from your standard terminal
\ and you'll be running independent
\ FORTH tasks from your second and third terminals connected to the UART module.

```

```

DECIMAL      \ compile this section in decimal base

\ NOTE: YOU MUST MAKE SURE THAT UART_MODULE_NUM CONSTANT CORRESPONDS TO YOUR HARDWARE!!
4 CONSTANT UART_MODULE_NUM  \ double check your hardware jumper settings!!!

\ default values used in Default_UART_Init (edit these to suit your requirements):
8          CONSTANT DEFAULT_BITS_PER_CHAR
1          CONSTANT DEFAULT_STOP_BITS
NO_PARITY  CONSTANT DEFAULT_PARITY
19200     CONSTANT DEFAULT_BAUDRATE
RS232     CONSTANT DEFAULT_PROTOCOL
FALSE     CONSTANT DEFAULT_MODEM_SUPPORT

: Default_UART_Init  ( module_num -- result )
  \ initializes BOTH channel1 and channel2 on the specified uart module_num.
  \ result = SUCCESS (=0) or BAD_PROTOCOL_COMBO (=1)
  \ this routine demonstrates how to initialize the uarts using default settings;
  \ the user should customize the parameters to suit the application.
  \ CAUTION: if decimal baud rates are hard-coded into this routine, make sure that
  \ this routine is compiled in decimal base.
  LOCALS{ &module }
  \ configure channel1:
  DEFAULT_BITS_PER_CHAR DEFAULT_STOP_BITS DEFAULT_PARITY CHANNEL1
  &module      ( numbits\numStopBits\parity_code\channel_num\module_num -- )
  Set_Data_Format      ( -- )
  DEFAULT_BAUDRATE CHANNEL1 &module      ( baud\channel_num\module_num -- )
  Set_Baud              ( -- )
  \ configure channel2:
  DEFAULT_BITS_PER_CHAR DEFAULT_STOP_BITS DEFAULT_PARITY CHANNEL2
  &module      ( numbits\numStopBits\parity_code\channel_num\module_num -- )
  Set_Data_Format      ( -- )
  DEFAULT_BAUDRATE CHANNEL2 &module      ( baud\channel_num\module_num -- )
  Set_Baud              ( -- )
  \ set protocols for each channel:
  DEFAULT_MODEM_SUPPORT DEFAULT_PROTOCOL DEFAULT_PROTOCOL
  &module      ( Ch1_modem_support\Ch1_protocol\Ch2_protocol\module_num -- )
  Set_Protocols        ( -- result )
;

HEX      \ variable area MUST be in common memory! ie., USE.PAGE, or HEX 8E00 0 VP X!
400 V.INSTANCE:  CH1_TASK  \ 1 Kbyte per task area
400 V.INSTANCE:  CH2_TASK  \ 1 Kbyte per task area

: CH1_Monitor  ( -- )      \ infinite task loop for CH1_TASK
  CFA.FOR CH1_EMIT UEMIT X!  \ revector this task's serial routines to use channel1
  CFA.FOR CH1_Ask_KEY U?KEY X!
  CFA.FOR CH1_KEY  UKEY X!
  CR ." Starting CH1_Monitor..."
  QUIT              \ call the infinite-loop FORTH monitor
;

: CH2_Monitor  ( -- )      \ infinite task loop for CH2_TASK
  CFA.FOR CH2_EMIT UEMIT X!  \ revector this task's serial routines to use channel2
  CFA.FOR CH2_Ask_KEY U?KEY X!
  CFA.FOR CH2_KEY  UKEY X!
  CR ." Starting CH2_Monitor..."
  QUIT              \ call the infinite-loop FORTH monitor
;

```

```
: Run_Demo ( -- )
  \ builds and activates two forth-monitor tasks,

  \ each using a separate channel on the uart module.
  UART_MODULE_NUM Set_UART_Number \ set global variable, must match hardware
  UART_MODULE_NUM Default_UART_Init ( -- result ) \ initialize the hardware
  IF CR ." Error: Invalid protocol combination was specified!" CR
  ELSE
    RELEASE.ALWAYS SERIAL.ACCESS ! \ ensure lots of PAUSES in Forth task
    (STATUS) NEXT.TASK ! \ empty the task loop
    0\0 0\0 0\0 CH1_TASK BUILD.STANDARD.TASK
    0\0 0\0 0\0 CH2_TASK BUILD.STANDARD.TASK
    CFA.FOR CH1_Monitor CH1_TASK ACTIVATE
    CFA.FOR CH2_Monitor CH2_TASK ACTIVATE
    START.TIMESLICER
  ENDIF
;

4 PAGE.TO.FLASH
5 PAGE.TO.FLASH
6 PAGE.TO.FLASH
STANDARD.MAP
SAVE
```

Hardware Schematics