

C Demonstration Program

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

```
// *****
// FILE NAME:   UModDemo.c
// 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 UART Module kernel extension file Install.txt
// MUST be loaded into memory 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
// tasks using the two UART Module serial ports.  Each task simply
// echoes incoming characters back to the terminal.
// When the top level function main() is running, the QED Board
// or Panel-Touch Controller is simultaneously using 3 serial ports:
// the standard primary serial port is running the QED interactive monitor,
// and each of the two serial channels on the UART Module is echoing characters.
// 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 C serial I/O routines such as
// putchar, puts, getchar, gets, printf, and scanf can be made to use
// any specified serial channel.  All that is required is to customize
// three functions named Key, AskKey, 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 // this constant MUST match hardware jumper settings!
// int Default_UART_Init( int module_num ) // demonstrates how to initialize module
// void main(void) // runs the demo program

// -----
// 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 or Panel-Touch Controllers with WildCard Carrier Board.

#include <mosaic\allqed.h> // include all of the qed and C utilities
#include "library.c" // this is a kernel extension file;
// assume it's in the same directory as this file; if not, edit the file specification

// ***** DEMONSTRATION PROGRAM *****
```

```

// The default task runs an interactive monitor as usual, using the 68HC11 UART.
// We create a second task and a third task that echo all incoming chars,
// each communicating using a serial channel on the UART Module.
// To run this demonstration, simply execute:
//     main
// You'll be running independent serial-echo tasks
// from your second and third terminals connected to the UART module.

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

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

// Define and allocate RAM for the task areas:
TASK ch1_task; // 1 Kbyte per task area
TASK ch2_task; // 1 Kbyte per task area

int Default_UART_Init( int module_num )
    // 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.
{
    // configure channel1:
    Set_Data_Format(DEFAULT_BITS_PER_CHAR,DEFAULT_STOP_BITS,DEFAULT_PARITY,1,module_num);
    Set_Baud(DEFAULT_BAUDRATE,1,module_num);
    // configure channel2:
    Set_Data_Format(DEFAULT_BITS_PER_CHAR,DEFAULT_STOP_BITS,DEFAULT_PARITY,2,module_num);
    Set_Baud(DEFAULT_BAUDRATE,2,module_num);
    // set protocols for each channel:
    return Set_Protocols(DEFAULT_MODEM_SUPPORT,DEFAULT_PROTOCOL,DEFAULT_PROTOCOL,module_num);
}

void CH1_Monitor(void)
    // infinite task loop for ch1_task, simply echoes all incoming chars on channel1
{
    uchar this_char;
    UKEY = (xaddr) CH1_KEY_XADDR; // defined in library.h
    UASK_KEY = (xaddr) CH1_ASK_KEY_XADDR; // defined in library.h
    UEMIT = (xaddr) CH1_EMIT_XADDR; // defined in library.h
    printf("Ready to echo incoming characters on Channel1...\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
    }
}

void CH2_Monitor(void)
    // infinite task loop for ch2_task, simply echoes all incoming chars on channel2
{
    uchar this_char = 0;
    UKEY = (xaddr) CH2_KEY_XADDR; // defined in library.h

```

```

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

```