# Microchip MLA TCP-IP modifications

Richard W. Wall

University of Idaho

December 4, 2013

The following modifications to the Microchip MLA 6/2013 version target the Cerebot MX7cK Processor board. The major revisions allow STACK\_USE\_UART to be defined in TCPIP ETH795.h and directs all UART communications to the UART1 as provided on the CEREbot MX7cK. This version of the TCPIP Stack also uses the PMP for the LCD interface provided by LCDlib.c and LCDlib.h instead of the bit-banging interface provided by LCDBlocking.c and LCDBlocking .h.

Project file changes (Changes that Dr. Wall made in creating IP\_PKG\_12-2013)

Under Headers/TCPIP Stack folder, removed LCDBlocking.h and add LCDlib.h

Under the Source/TCPIP Stack folder, removed LCDBlocking.h and add LCDlib.h

Changes to the Microchip MLA library

Microchip\TCPIP Stack\UART.c - Total rewrite

Microchip\Include\TCPIP Stack\TCPIP.h - Total rewrite for UART1

Microchip\TCPIP Stack\Delay.c - Total rewrite

Microchip\Include\TCPIP Stack\TCPIP.h - Modified for using LCDlib (PMP 8 bit control)

Microchip\TCPIP Stack\LCDlib.c - New file

Microchip\Include\TCPIP Stack\LCDlib.c - New file

Application Files

application/configs/HardwareProfile.h

Add: #define LCD\_PMP in HardwareProfile.h

Commented out all references to bit-banging LCD

/\* These definitions are used for bit-banging LCD control in LCDblocking.h

#define LCD\_MASK 0x00ff

#define LCD\_DATA\_TRIS (TRISE)

#define LCD\_DATA\_OUT() TRISECLR = LCD\_MASK

#define LCD\_DATA\_IO (LATE)

#define LCD\_RD\_WR\_TRIS (TRISDbits.TRISD5)

#define LCD\_RD\_WR\_IO (LATDbits.LATD5)

#define LCD\_RS\_TRIS (TRISBbits.TRISB15)

#define LCD\_RS\_IO (LATBbits.LATB15)

#define LCD\_E\_TRIS (TRISDbits.TRISD4)

#define LCD\_E\_IO (LATDbits.LATD4)

\*/

Setup UART1

// UART configuration (not too important since we don't have a UART

// connector attached normally, but needed to compile if the STACK\_USE\_UART

// or STACK\_USE\_UART2TCP\_BRIDGE features are enabled.

#define UARTTX\_TRIS (TRISFbits.TRISF8)

#define UARTRX\_TRIS (TRISFbits.TRISF2)

// UART mapping functions for consistent API names across 8-bit and 16 or

// 32 bit compilers. For simplicity, everything will use "UART" instead

// of USART/EUSART/etc.

#define BusyUART() BusyUART1()

#define CloseUART() CloseUART1()

#define ConfigIntUART(a) ConfigIntUART1(a)

#define DataRdyUART() DataRdyUART1()

#define OpenUART(a,b,c) OpenUART1(a,b,c)

#define ReadUART() ReadUART1()

#define WriteUART(a) WriteUART1(a)

#define getsUART(a,b,c) getsUART1(b,a,c)

#define putsUART(a) putsUART1(a)

#define getcUART() ReadUART1()

#define putcUART(a) WriteUART1(a)

#define putrsUART(a) putsU1(a)