

Microchip PIC32 Reference Designs

Dr. Richard Wall – Professor
Department of Electrical and Computer Engineering
University of Idaho
Moscow, ID 83844-1023
January 3, 2013
rwall@uidaho.edu

I. General Notes:

This collection of 32 reference designs specifically the Microchip PIC32MX795F512L microcontroller and the Digilent Cerebot MX7ck processor board. Most of the projects are based on the PIC32 port of FreeRTOS and the Microchip Application Libraries. There was no cost for the software downloaded from the web although users of the FreeRTOS and Microchip must be adhered to licensing agreements for the software that they generated. Microprocessor code for which I am the author, I place into the public domain. This code is provided “as is” and is without warranty of any kind either expressed or implied. This purpose of the reference designs and users guides is strictly educational. I wish to thank Digilent Inc. of Pullman WA for providing the hardware that enabled me to develop these reference designs. I will be grateful for any feedback, corrections, and suggestions users and particular fellow educators may have regarding the educational attributes of the reference designs.

The reference designs are grouped into three primary areas of emphasis: real time operating systems, Ethernet control and instrumentation, and mesh networks. The real time operations reference design use principles of multi tasking based upon cooperation, time sharing and priority preemption. TCP/IP Ethernet control focuses on man-machine interface capabilities using the PIC32 as a web server in conjunction with common PC web browsers. UDP Ethernet is used to implement machine to machine interfaces. When the processor has access to the internet, ICMP/IP is used for automatic email generation. The last area of emphasis is mesh networks. The Microchip P2P and MiWi protocols are based on the IEEE 802.15.2 standard. These two mesh network protocols are similar to but not exact implementations to the ZigBee that targets low cost low power sensor network applications.

Richard Wall

- A. Hardware Platform:** The list of reference designs and examples were developed for the Cerebot PIC32 MX7ck processor board an associated peripheral modules shown in Figure 1. An alternate configuration utilizing the Cerebot PIC32MX7 (depreciated) is shown in Appendix I. If the UART is used as a console, the PIC32 UART1 is used at 19200 BAUD. All project support the PmodCLP LCD module. Some projects also support a 4X20 character LCD using a four data bit mode that interfaces to 7 pins of PORT E through connection to JB. Additional details for this interface are provided in the appendix. Information on all hardware modules is available at <http://www.digilentinc.com/>. The 32 KHz, X2, crystal must be populated on the processor board in order for reference designs that use the RTCC to function.

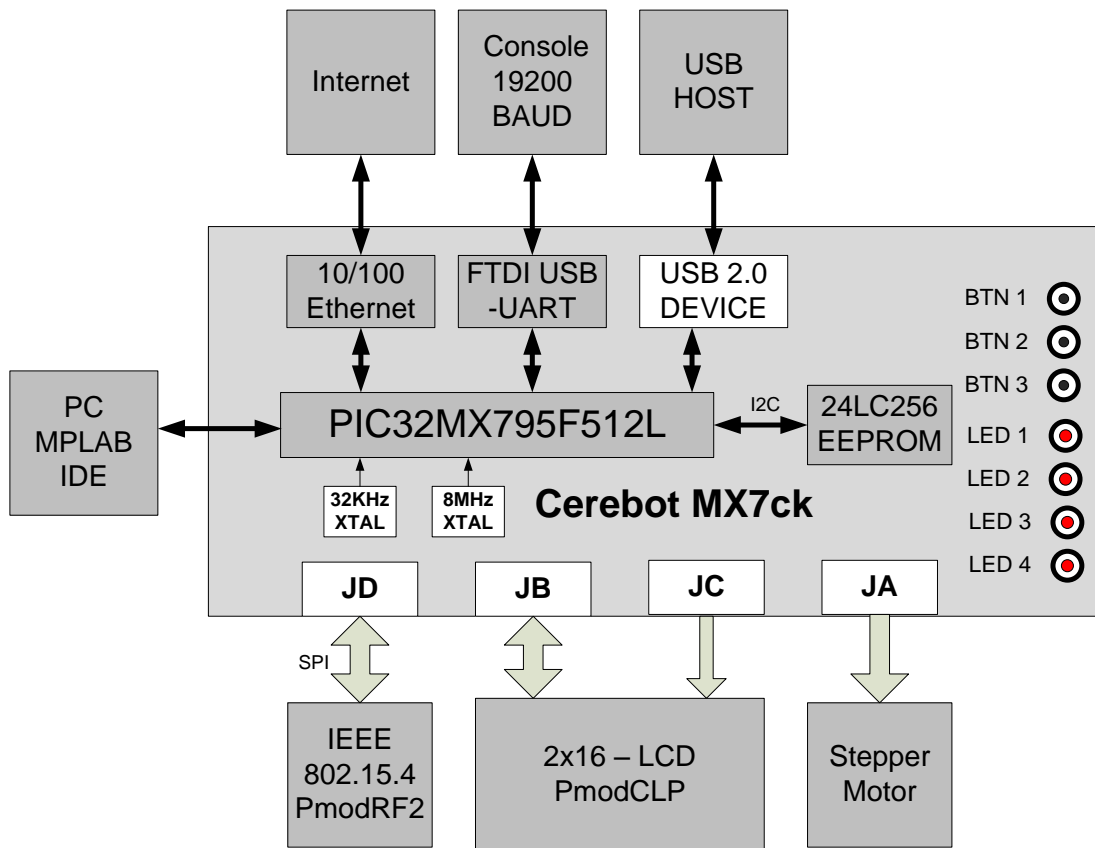


Figure 1. Cerebot PIC32 MX7ck Development Platform

B. Software Environment:

The designs were tested using MBLAB on both versions 8.86 and 8.88 with the Microchip MPLAB C32 C compiler version 2.02.

Reference Designs involving the Ethernet and MiWi or P2P examples are based upon the Microchip Applications Library (MAL) v2012-10-15. See

http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2680&dDocName=en547784

1. Software Customization

The latest revision of C32, MPLAB and MAL, if different that those listed, may require software modifications. The directory structure illustrated in Figure 2 is used for all of the accompanying reference designs whether the MAL is used or not. In my opinion, the MAL directory structure is not organized in the most optimal manner but I have learned to navigate through it with a degree of trial and error. My advice for building new projects is to copy an existing reference design to a new folder and modify, add, and delete code as needed. Additional projects should be added at the “Application Projects” layer or the “Project X” layer to maintain the organization structure. Additional directories may exist but are not included under the Microchip MAL because they are not needed by the current reference designs.

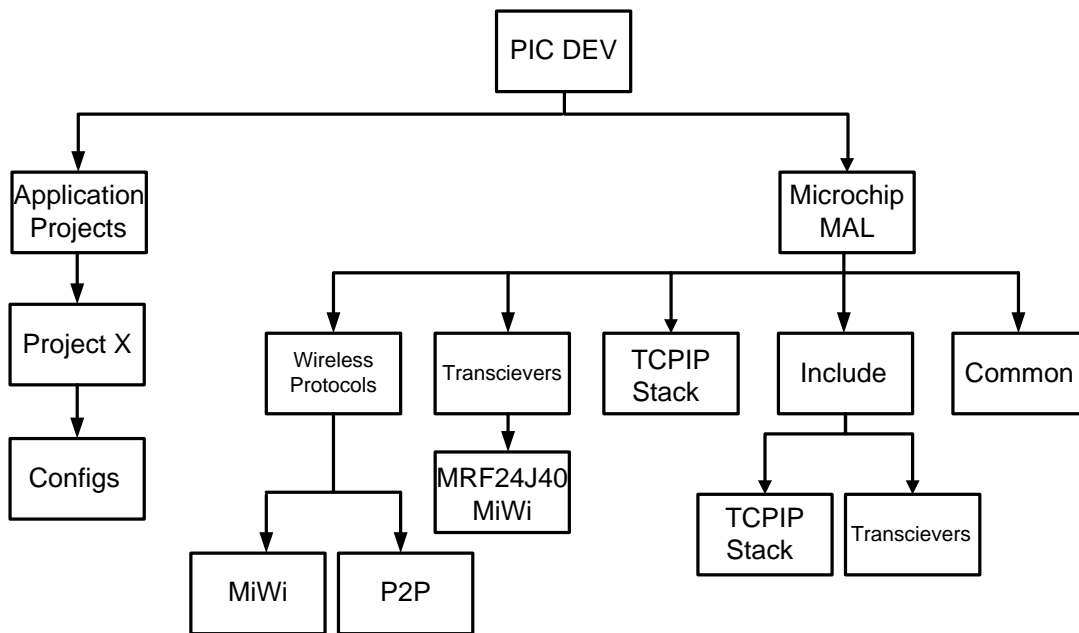


Figure 2. Software development environment architecture

The following list summarizes the modifications to the MAL to suit my purposes and to accommodate the Digilent Cerebot MX7ck / Cerebot 32MX7 (obsolete) processor boards as well as the other attached hardware.

- a. Timers and Delays:
 - i. Delay.c in the MAL Delay function has been modified to utilize the core timer to the time base for the DelayMs, DelayUs, and Delay10us functions. The Delay.h file that must be referenced to use the delay functions is in the TCPIP STACK Directory under the Include Directory.
 - ii. TimeDelay.c Is a function identical to Delay.c. This file resides in the Microchip/Common directory and is used with MiWi applications.
- b. LCD drivers
 - i. The LCDblocking function required timing adjustments for the bit-banging handshaking. The LCD operations are changed to my own LCD routines for most reference designs. Only reference designs using the WebVend tutorials use a modified version of the LCDblocking function that was originally supplied with the MAL.
 - ii. All other reference designs use the LCD8bit and LCD4bit LCD driver files that I have written for the specific hardware environment. These files are kept in a subdirectory “Common” and “Include” that exist under the Microchip directory. If your hardware environment uses the Digilent PmofCLP LCD display, comment out the line of code, `#define LCD_4BIT_4LINE`, in HardwareProfiles.h file that is kept in the Configs subdirectory under the specific project directory.
- c. NVM and I2C_EEPROM drivers provide the ability to store application constants that can be loaded into the application at power up. This nonvolatile memory uses the 23L256 EEPROM on the Digilent Cerebot PIC32MX7 processor boards. I2C_EEPROM is the driver software for 24LC256 hardware and NVM uses. The MAL EEPROM.c is not used for any reference designs.
- d. NIST.c is an application program that is used with some of the Ethernet Reference designs. The purpose of this program is to set the PIC32 real time clock and calendar (RTCC) based upon a NIST time service.

The 32KHz crystal (X2) must be installed and the secondary oscillator enabled in order to use the RTCC applications. Refer to RD18 through RD20 for additional details.

- e. TimeDelay.c is identical to Delay.c but is used with some of the MiWi and P2P reference designs to provide a degree of independence from the TCPIP STACK applications.

2. Directory Structure

There are numerous reference designs that are based around a common theme. The succession of reference designs is to add complexity in small steps that is representative of the spiral design methodology. I have found these particularly useful when introducing new concepts in an academic environment.

For reference designs involving real-time operating systems one should refer to the help documents provided at <http://www.freertos.org/>. FreeRTOS applications do not require support code in the MAL unless integrating a FreeRTOS application with either and TCPIP Stack or MiWi wireless application. To use FrerERTOS to manage a MiWi or TCPIP Stack application the directory labeled “My RTOS Directory” shown in Figure 3 should be places at the same layer as “Application Projects” shown in Figure 2.

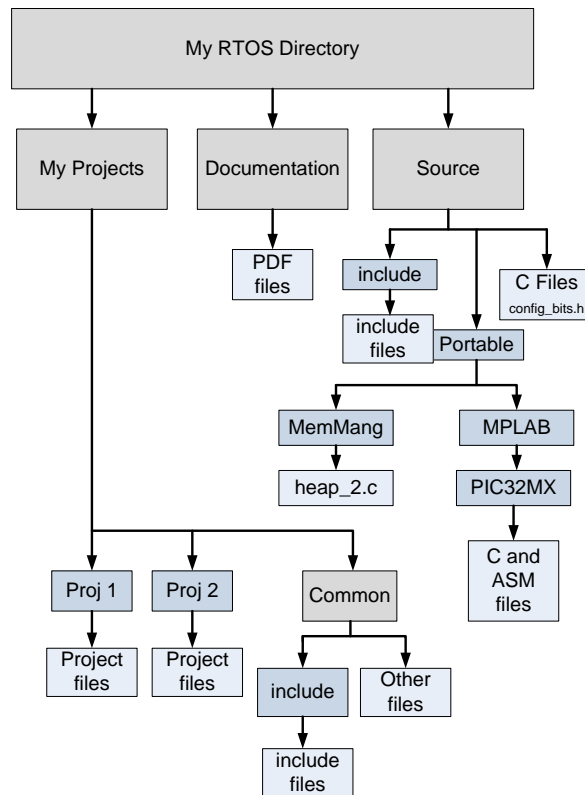


Figure 3. FreeRTOS project development directory structure

C. Design Tutorials and Reference Designs:

You will most likely be prompted to specify the location of the compiler or linker the first time the project is recompiled on your PC. Each of the links below directs you to a information guide and description of targeted reference designs.

[FreeRTOS Reference Designs](#)

[TCP/IP Reference Designs](#)

[P2P and MiWi Reference Designs](#)

[Reference Design Code](#)

Appendix I

Hardware configuration for Cerebot PIC32MX7 Processor Board

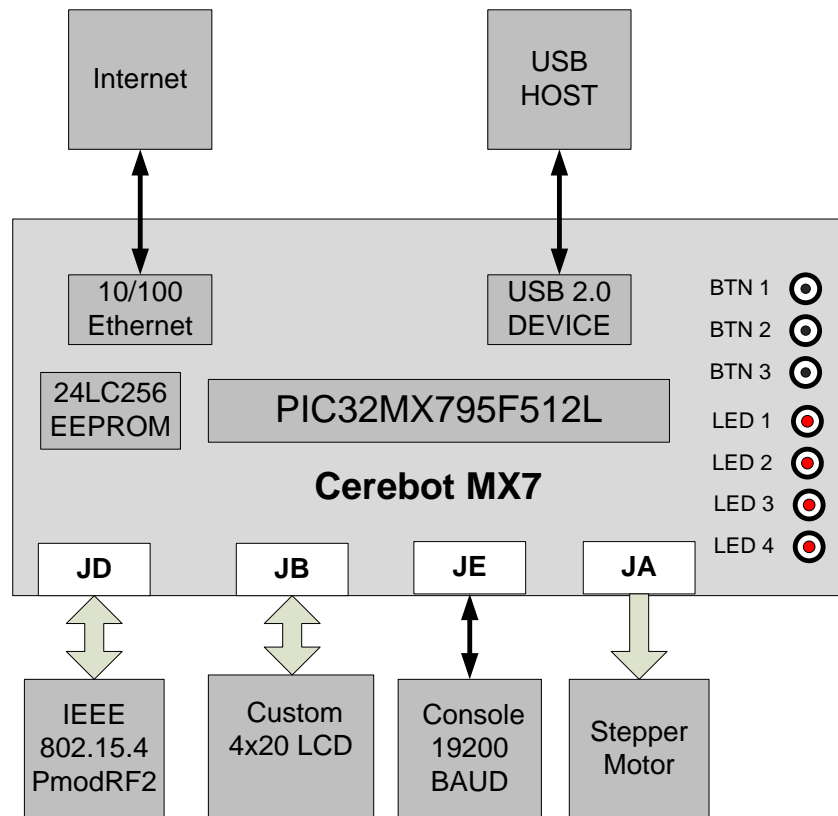


Figure 4. Cerebot PIC32 (deprecated) hardware development platform

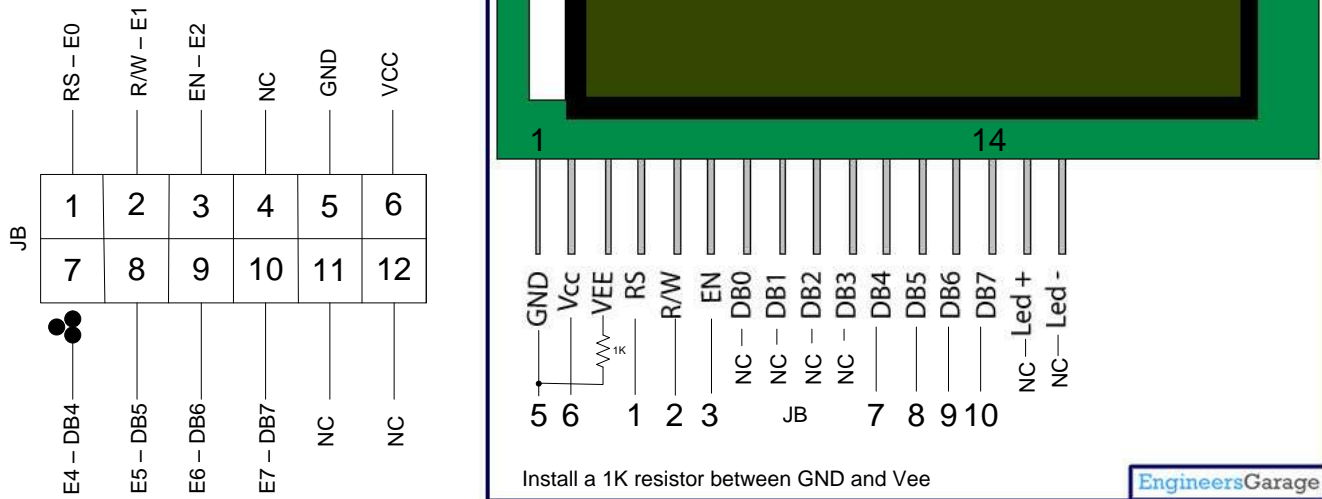


Figure 5. Custom interface connections for 4x20 LCD using 4-bit interface. VCC must be set for 5VDC.

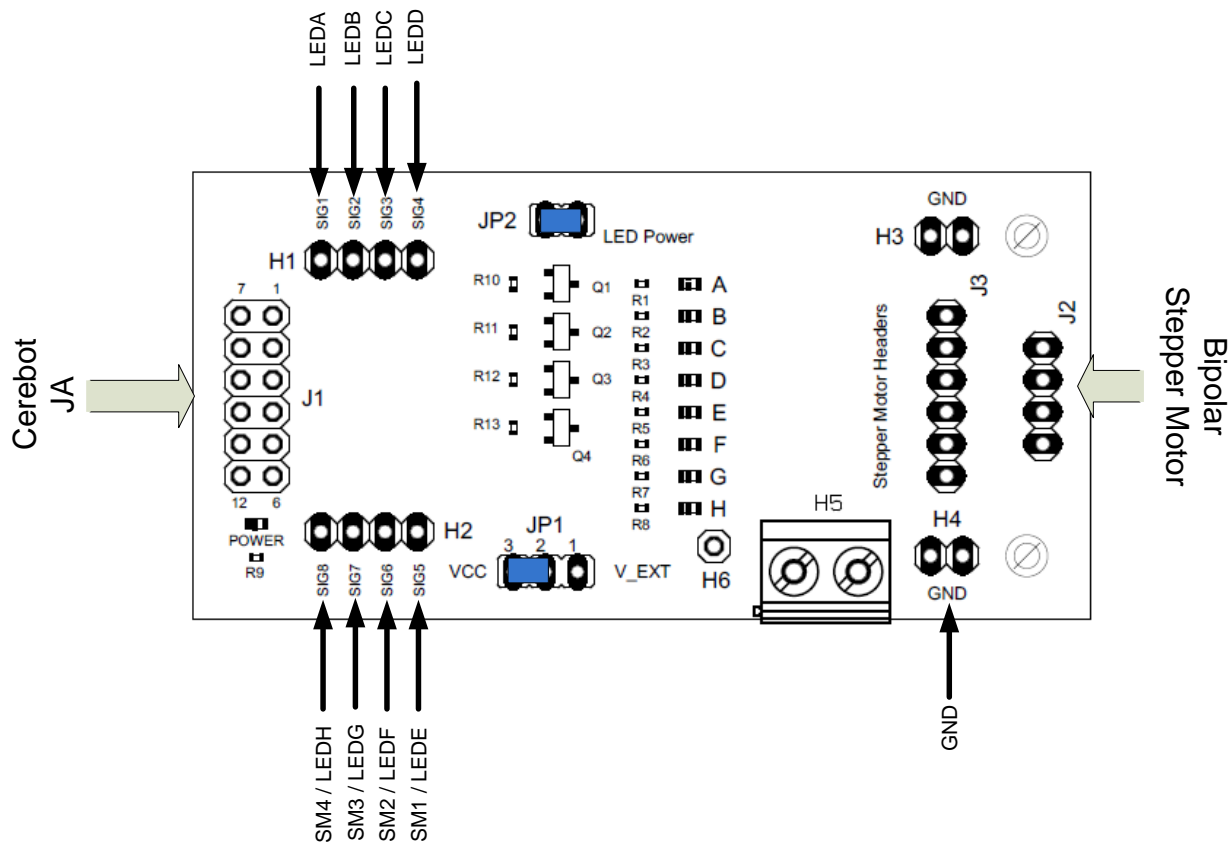


Figure 6. Stepper Motor Driver Module parts layout. VCC should be set for 5VDC.