Published on *Analog Digital Lab:

electronics articles and tutorials* (http://10.240.11.2)

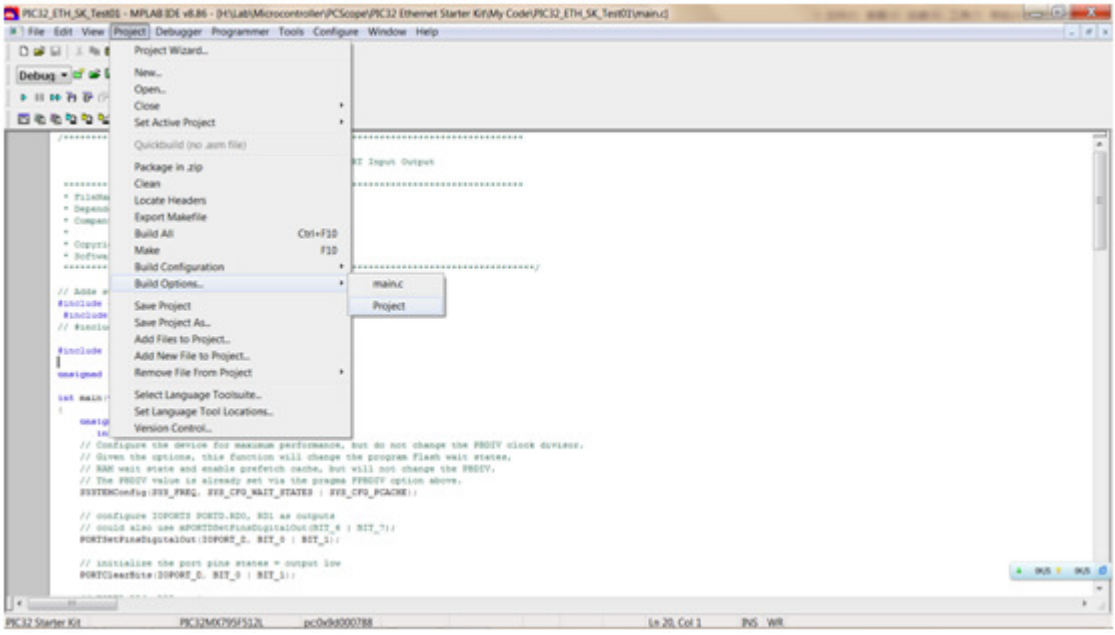
posted by maintenance on Wed, 02/06/2013 - 13:54

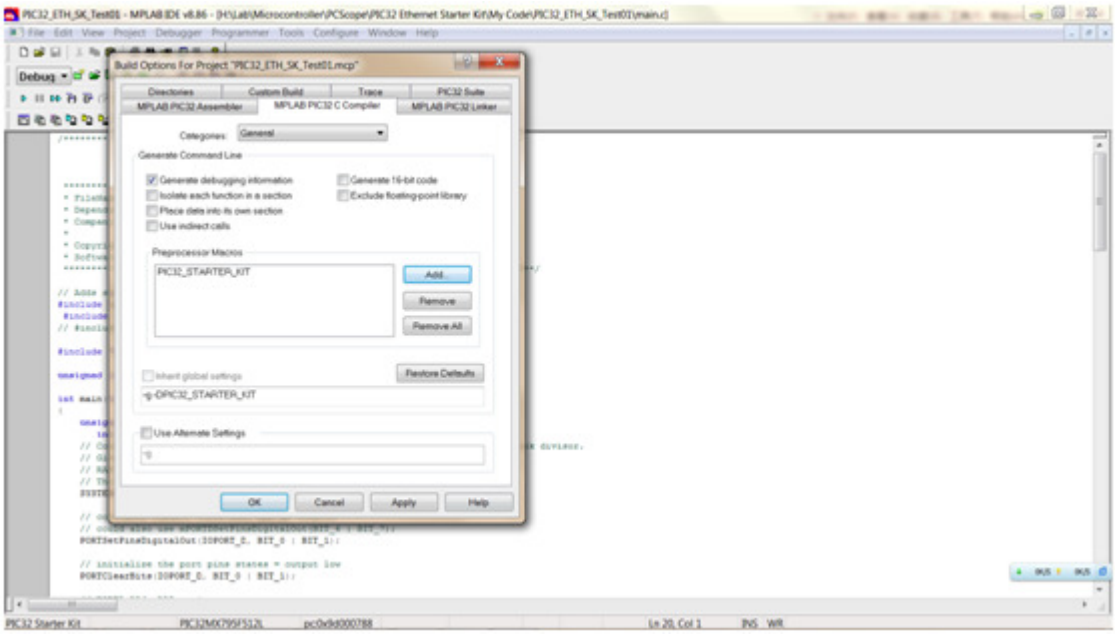
PIC32: Ethernet Starter Kit TCP Client

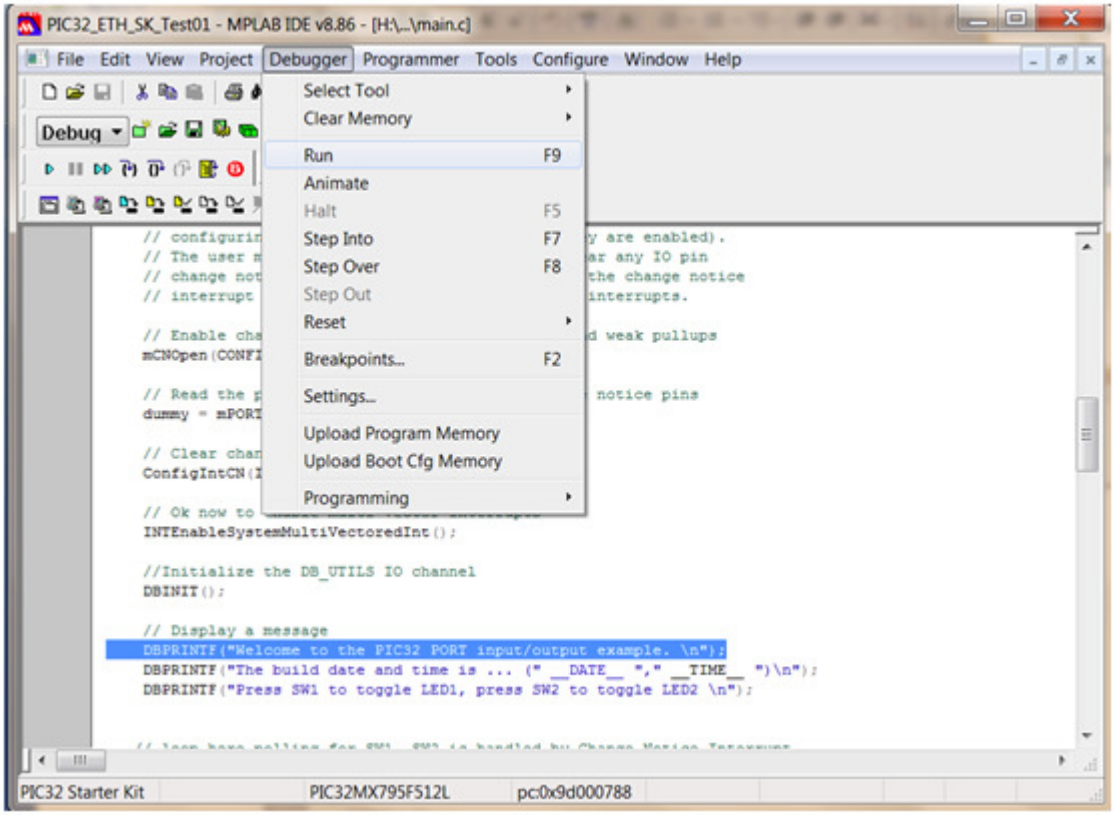
Set debuger "DBPRINTF"

You might wonder why it didn't show any debug text when using e.g. *DBPRINTF("Welcome to the PIC32 PORT input/output example. \n").* Well, this is because you still need to make some adjustment.

Project->Build Options->Project->C-compiler->Macros->add...-> "PIC32_STARTER_KIT"



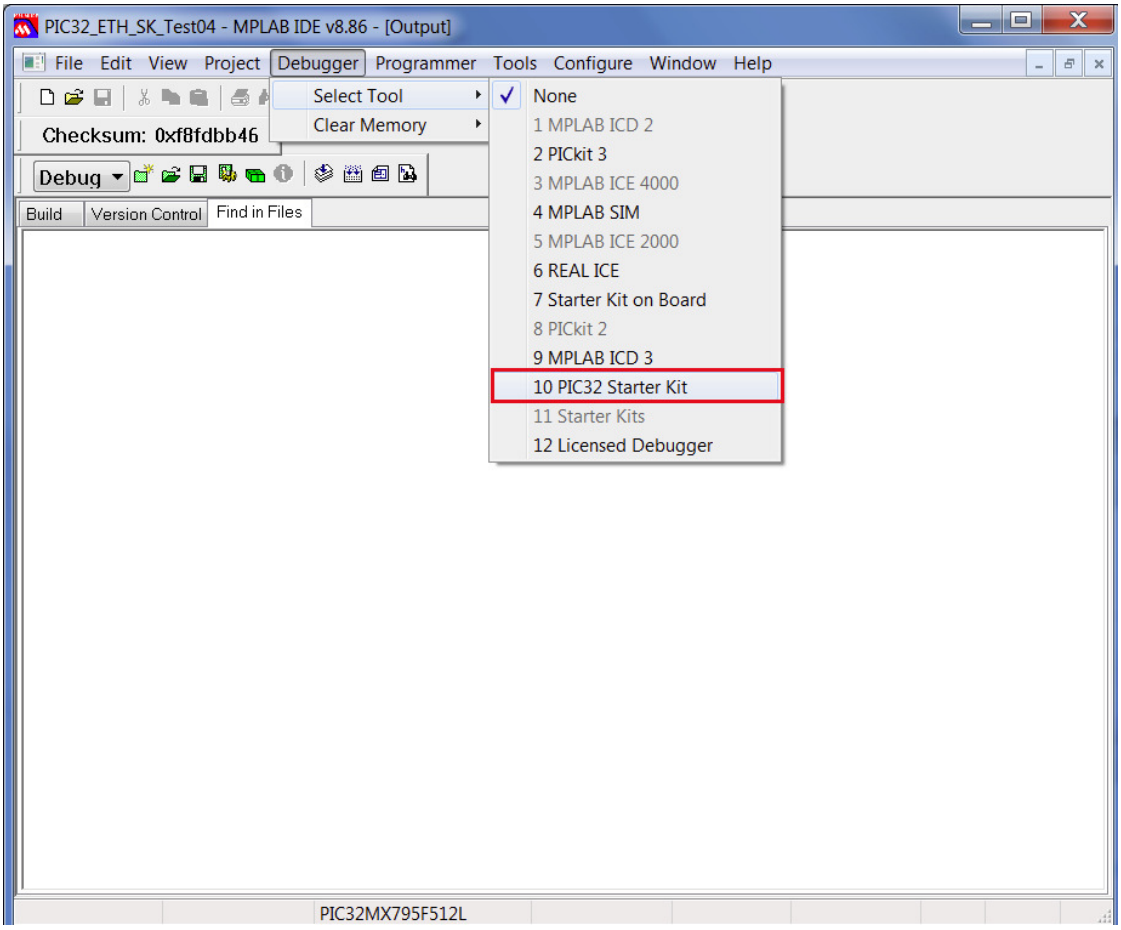


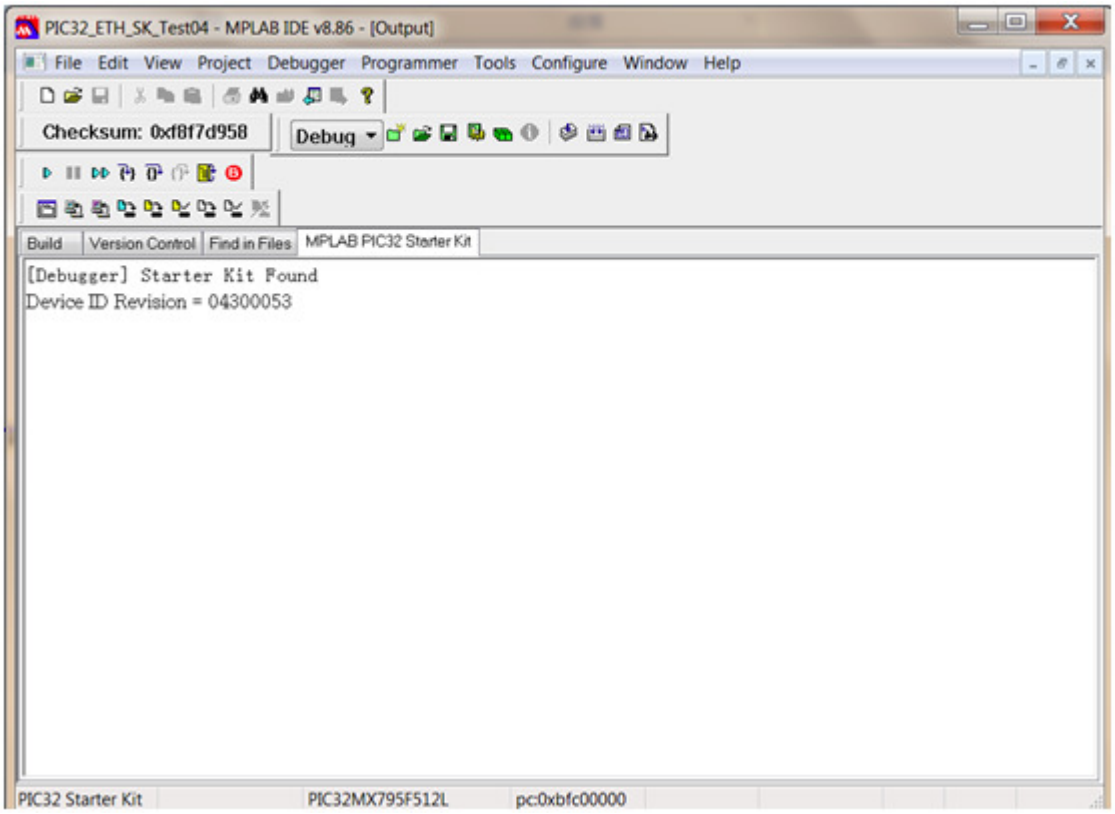


By following these steps you will be able to see the debug text.

How to switch between debugging and use states?

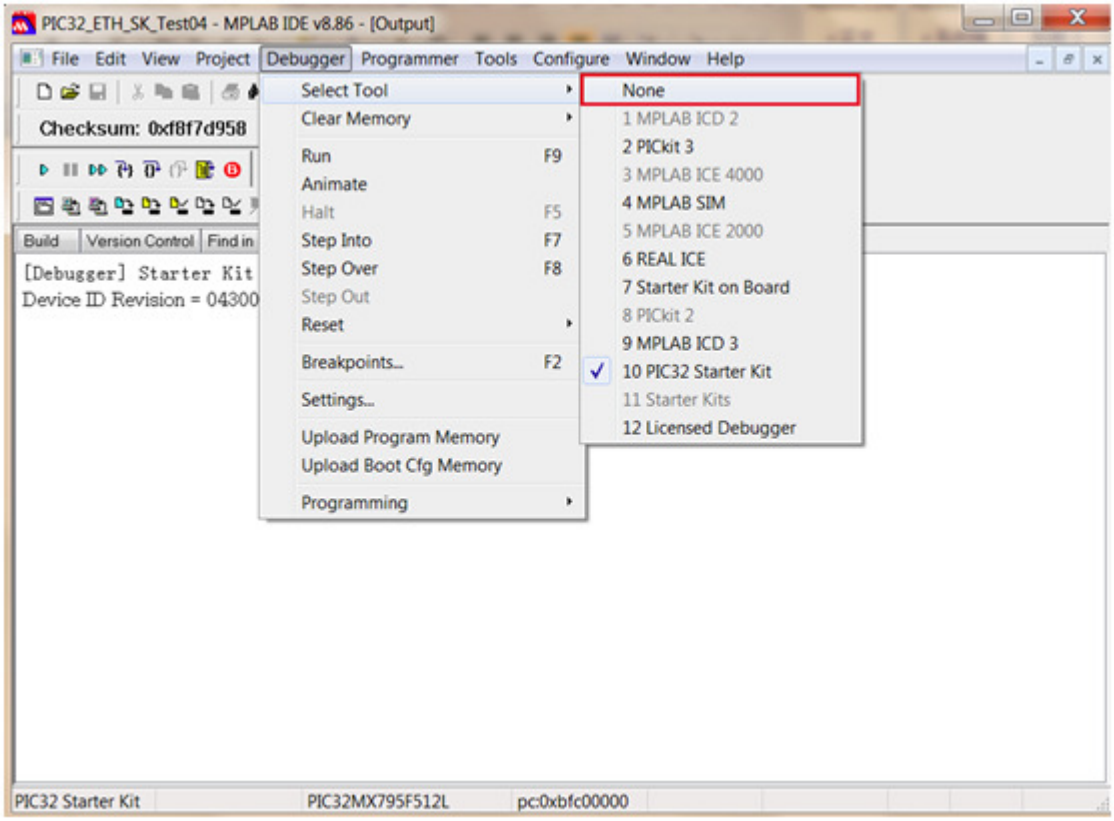
1. To connect the ESK board

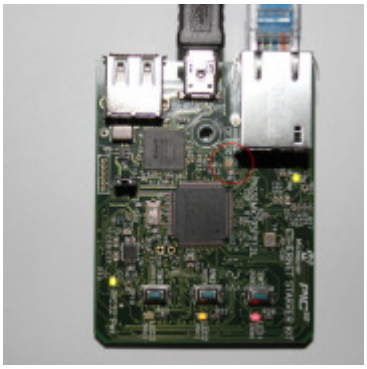

PIC32_ETH_SK_Test04 - MPLAB IDE v8.86 - [Output]
File Edit View Project Debugger Programmer Tools Configure Window Help
Select Tool
Clear Memory
None
1 MPLAB ICD 2
2 PICkit 3
3 MPLAB ICE 4000
4 MPLAB SIM
5 MPLAB ICE 2000
6 REAL ICE
7 Starter Kit on Board
8 PICkit 2
9 MPLAB ICD 3
10 PIC32 Starter Kit
11 Starter Kits
12 Licensed Debugger
Checksum: 0xf8fdbb46
Debug
Build
Version Control
Find in Files
PIC32MX795F512L


PIC32_ETH_SK_Test04 - MPLAB IDE v8.86 - [Output]
File Edit View Project Debugger Programmer Tools Configure Window Help
Checksum: 0xf8f7d958
Debug
Build
Version Control
Find in Files
MPLAB PIC32 Starter Kit
[Debugger] Starter Kit Found
Device ID Revision = 04300053
PIC32 Starter Kit
PIC32MX795F512L
pc:0xbfc00000

(Click to enlarge)

2. Disconnect the board then the ESK board can run the loaded program





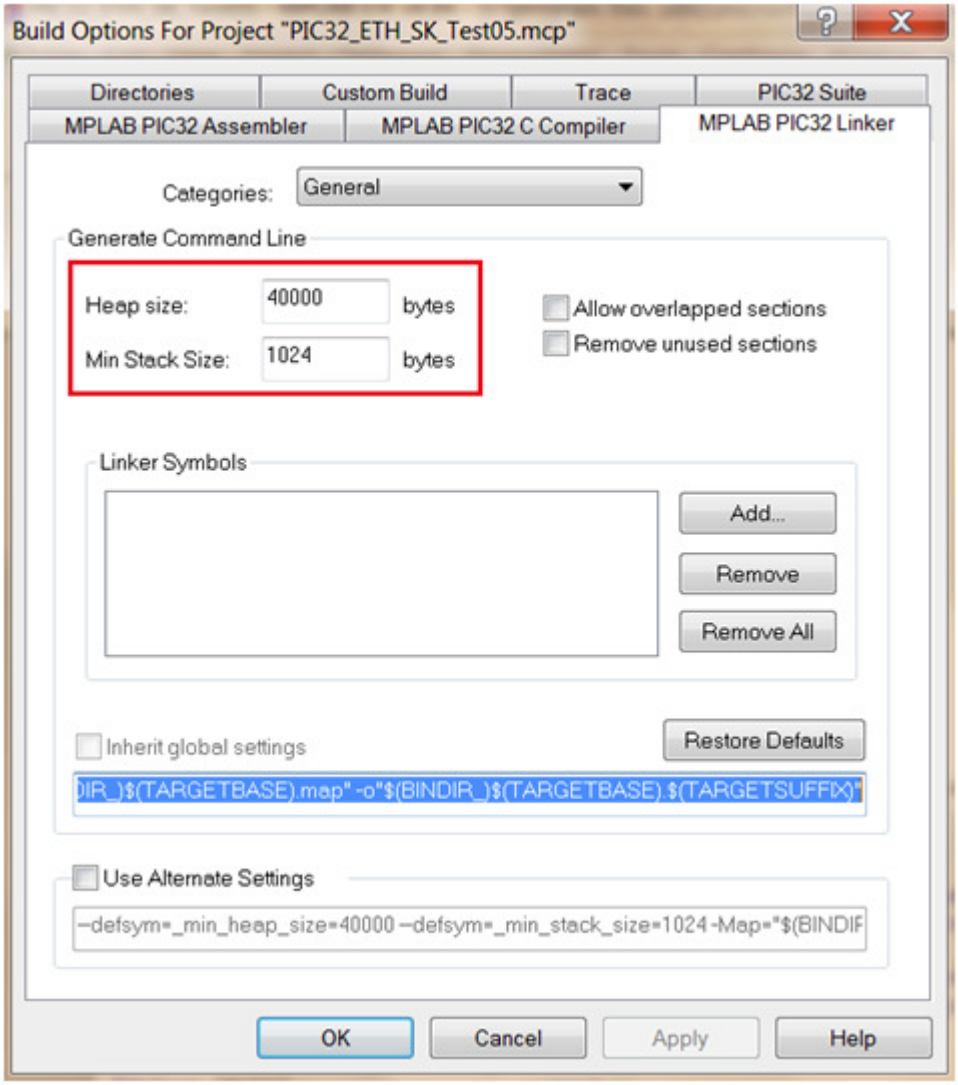
(Click to enlarge)

3. Reconnect the board again and the board will switch to debug mode then you can program the board again

Why DHCP doesn't work?

The "TCPIP-BSD HTTP Server" demo has the DHCP client disabled by default. To enable it the symbol DEMO_USE_DHCP=0 should be replaced with DEMO_USE_DHCP=1 under "MPLAB PIC32 C Compiler" settings in the "Build Options" for the project(similar to "DBPRINTF" setting).

Other settings for Ethernet



Send large amount of data to PC via TCP

TCP is a stream based service, unlike UDP where you send a packet at a time, with TCP the stack receives "enough" data to fill a payload or has been waiting to send data in a buffer for some pre-defined period. The stack will have an MTU defined (maximum transmission unit) which basically means the maximum size, including all headers, of a packet. With TCP, the stack must retain a copy of any transmitted data in a retransmit buffer until the data has been acknowledged. The acknowledged data is then removed from the retransmit buffer. If the data is not acknowledged, for example a packet is lost, a timer on the transmit side will "pop" and the stack will retransmit the unacknowledged data in the retransmit buffer.

PICs have relatively limited amounts of RAM available for maintaining large retransmission buffers and large frame sizes. The Ethernet controller has some buffer memory available, not enough to be really useful but enough for most stack applications. The stack maintains a maximum window size which is used to indicate how much free space the receiver has available to receive data from the transmitter. This mechanism is used by both ends of a TCP connection.

When configuring the TCPIP stack, you carve up the Ethernet controllers memory into receive and transmit buffers. The transmit buffer may also contain the retransmit buffer. If you configure the Ethernet buffer pointers incorrectly then it is possible to corrupt the transmit and/or receive buffer contents in the Ethernet controller.

Ok, enough theoretical anlysis, let's take a look how this configuration is precisely done in a project when large amount of data is required.

In your TCPIPConfig.h:

```
/* TCP Socket Memory Allocation
 *   TCP needs memory to buffer incoming and outgoing data.  The
 *   amount and medium of storage can be allocated on a per-socket
 *   basis using the example below as a guide.
 */
   // Allocate how much total RAM (in bytes) you want to allocate
   // for use by your TCP TCBs, RX FIFOs, and TX FIFOs.
   #define TCP_ETH_RAM_SIZE              (0ul)
   #define TCP_PIC_RAM_SIZE              (16384ul)
   #define TCP_SPI_RAM_SIZE              (0ul)
   #define TCP_SPI_RAM_BASE_ADDRESS      (0x00)
```

Make sure you have allocated enough memory to your TCP socket, e.g. 16384ul

```
#define TCP_CONFIGURATION
      ROM struct
      {
         BYTE vSocketPurpose;
         BYTE vMemoryMedium;
         WORD wTXBufferSize;
         WORD wRXBufferSize;
      } TCPSocketInitializer[] =
      {
         {TCP_PURPOSE_GENERIC_TCP_CLIENT, TCP_PIC_RAM, 4000, 200},
         //{TCP_PURPOSE_GENERIC_TCP_SERVER, TCP_PIC_RAM, 20, 20},
         //{TCP_PURPOSE_TELNET, TCP_PIC_RAM, 200, 150},
```

```
    //{TCP_PURPOSE_TELNET, TCP_PIC_RAM, 200, 150},
    //{TCP_PURPOSE_TELNET, TCP_PIC_RAM, 200, 150},
    //{TCP_PURPOSE_FTP_COMMAND, TCP_PIC_RAM, 100, 40},
    //{TCP_PURPOSE_FTP_DATA, TCP_PIC_RAM, 0, 128},
    {TCP_PURPOSE_TCP_PERFORMANCE_TX, TCP_PIC_RAM, 2000, 1},
    {TCP_PURPOSE_TCP_PERFORMANCE_RX, TCP_PIC_RAM, 40, 2000},
    //{TCP_PURPOSE_UART_2_TCP_BRIDGE, TCP_PIC_RAM, 256, 256},
    {TCP_PURPOSE_HTTP_SERVER, TCP_PIC_RAM, 1500, 1500},
    {TCP_PURPOSE_HTTP_SERVER, TCP_PIC_RAM, 1500, 1500},
    {TCP_PURPOSE_DEFAULT, TCP_PIC_RAM, 1000, 1000},
    //{TCP_PURPOSE_BERKELEY_SERVER, TCP_PIC_RAM, 25, 20},
    //{TCP_PURPOSE_BERKELEY_SERVER, TCP_PIC_RAM, 25, 20},
    //{TCP_PURPOSE_BERKELEY_SERVER, TCP_PIC_RAM, 25, 20},
    //{TCP_PURPOSE_BERKELEY_CLIENT, TCP_PIC_RAM, 125, 100},
};
#define END_OF_TCP_CONFIGURATION
```

In my project, I need to send large amount of data from PIC to PC via TCP connection, therefore i have enlarged the wTXBufferSize to 4000, and since the PIC only needs to receive commands from the PC, a small quantity is chosed here for wRXBufferSize.

Source files download:

PIC32_ETH_SK_Test04.zip

Demo: Generic TCP Client

set HD 1080p quality, then you can see the code in the demo

PIC32 Ethernet Starter Kit DM320004 - Generic TCP Cl...

Tags:
microchip

PIC32
Ethernet Starter Kit
TCP
client

Comments

pic32 ethernet stater kit

Permalink On **Wed, 04/09/2014 - 13:44**

dear friends
i got stuck in incorrectly allocated your TCP socket buffers in Ethernet stater kit,please help me out
regards
sidharth rath
mail-rath.sidh@gmail.com

By **sidharth (not verified)**