PIC-MAXI-WEB MANUAL

Introduction

-PIC-MAXI-WEB board uses C18 compiler, the codes are very similar to C language
- The maximum memory that the chip can hold is 0x1FFFF
- Baud rate = 19200 bps
- Board Features:
    - PIC18F97J60 microcontroller with embedded Ethernet MAC and PHY and 128KB Flash programming space
    - 1Mbit on board serial flash for web pages storage
    - ICSP/ICD connector for programming and debugging with PICICD2 and PIC-ICD2-POCKET.
    - Temperature sensor
    - Trimmer potentiometer connected to analogue input
    - 2 relays 10A/250VAC
    - 4 optoisolated digital inputs
    - 12 analogue inputs on terminal block
    - Two buttons
    - LCD16x2 with backlight
    - RS232 interface
    - Ethernet interface
    - Complete web server and TCP-IP stack support as per Microchip's open source TCP-IP stack
    - Terminal block for power supply works with 9-25 VDC power supply
    - UEXT Extension header to connect to other boards

For more information, look under **PIC-MAXI-WEB Development Board.pdf**

Beginner

1.0 Installation

1) Install MPLAB IDE

http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en019469&part=SW007002

2) Install TCP/IP STACK

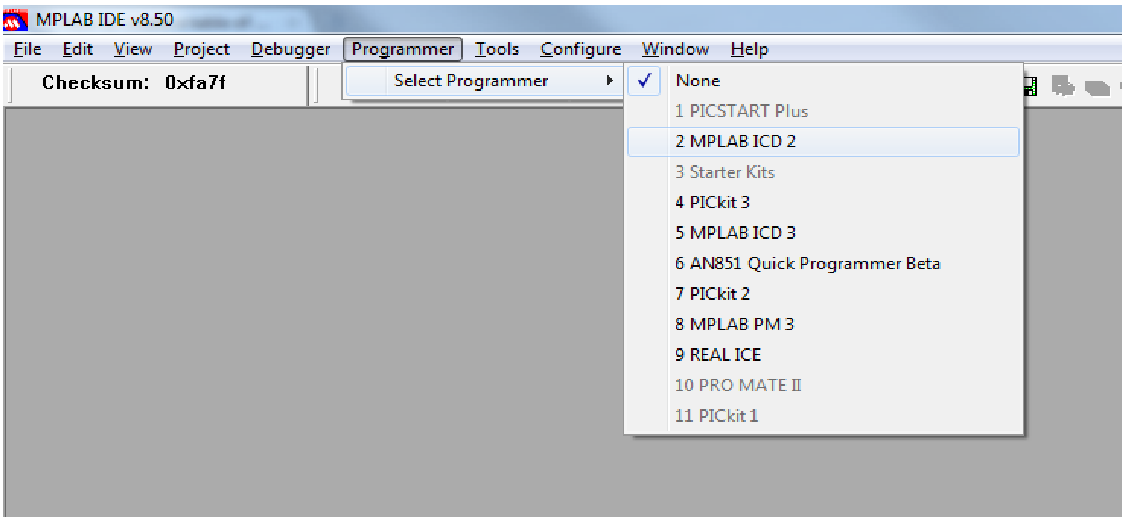
http://www.olimex.com/dev/pic-maxi-web.html

Note: You need to download TCP/IP stack v5.00 from Microchip's site. Olimex supplies only a patch due to user agreement issues. Please read the instructions in the patch package on how to apply the patch.

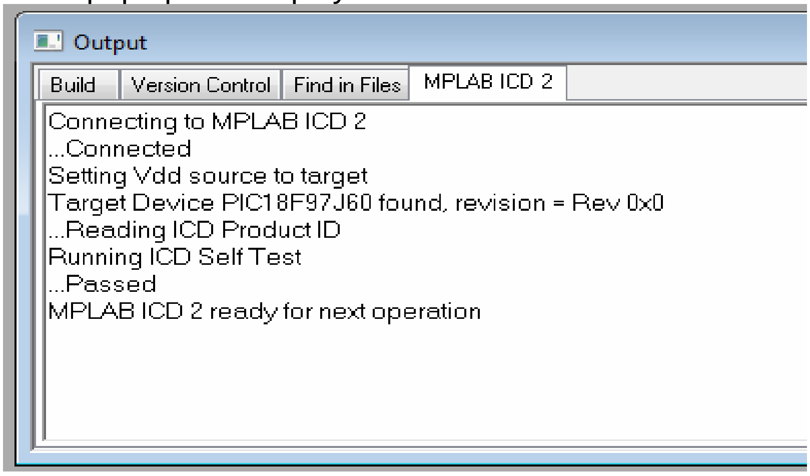
2.0 Setup

1) Using PIC-ICD2-POCKET, connect to board and computer before opening software, install as ICD2 if haven't installed the device before.
2) Open MPLAB IDE:

Programmer → Select Programmer → MPLAB ICD2



A window should pop up and display:



```
Connecting to MPLAB ICD 2
...Connected
Setting Vdd source to target
Target Device PIC18F97J60 found, revision = Rev 0x0
...Reading ICD Product ID
Running ICD Self Test
...Passed
MPLAB ICD 2 ready for next operation
```

3) Backup file from board (optional):

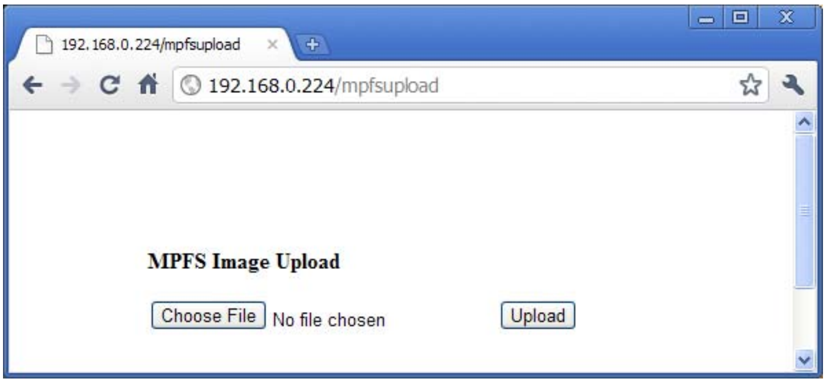
Programmer → Read, File → Export and save the .HEX file

4) Writing a prebuilt image (**also used to restore factory state of the board**)

Locate the **prebuilt** folder inside the patch package. With ICD2 already connected select File>Import... and select the file 'TCPIP Demo App-C18.hex' in the above mentioned folder. Then Program the board.

Next you will need to load the image file for the webpage. Connect a LAN cable, wait a bit and check the IP address displayed on the LCD. Open your favorite browser and enter the following (for example the IP is 192.168.0.224):

http://192.168.0.224/mpfsupload



192.168.0.224/mpfsupload

MPFS Image Upload

No file chosen

Press the button 'Choose file' and select the file 'MPFSImg2.bin' inside the 'prebuilt' folder. Then press 'Upload' and wait patiently for the upload to finish.

3.0 New Project & Programming Startup

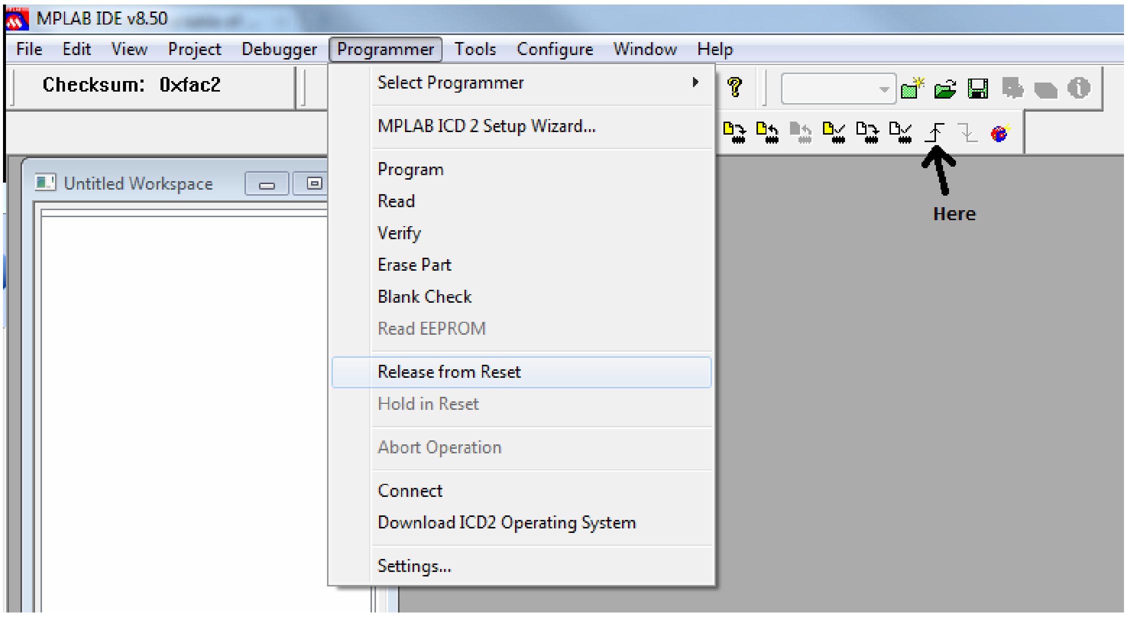
1) From Local C Drive, C:\svn\Development\Olimex
    - Copy New Starter Project Kit folder and rename it
    - Inside are the only two folders needed for any modifications
    - ***IMPORTANT NOTE*** Do not ever change the name of these two folders
2) Open MPLAB IDE:
    Project → Open… →TCPIP Demo App → TCPIP Demo App C18
3) Do any modification needed

4.0 Building Project

After applying the patch open the file 'TCPIP Demo App-C18.mcw' in folder 'TCPIP Demo App' with MPLAB. Please note that other project files inside this folder and in the other folders are part of the TCP/IP stack released by Microchip, but are not supported demos for PIC-MAXI-WEB.

1) Project → Build All
2) Programmer → Program
3) To run the program, Programmer → Release from Reset or the icon

See if everything goes as expected. Now you can do some modifications to the code as desired.



Advanced

Source files Description:

MainDemo.c
- Initialize LCD status, registers, Ethernet driver

CustomHTTPApp.c
- Modification for web server

TempsNPots.c
- Initialize Analog Inputs & display values onto the web server *ONLY*
- Any changes made to the temperature calculation, please also do the same change to the GenericTCPClient.c & MainDemo.c

GenericTCPClient.c
- Sends data packets to the specify port with time & date

GenericTCPServer.c
- Listen to the PC104 heartbeat status

TCPIPConfig.h
- Enable/Disable additional features that are already pre-programmed

HardwareProfile.h

- Within this file contains pre-defined register names, any additional registers that are used should be added onto the list first
- *ONLY* edit the list between the 2 rows of stars

```
//****************************************************************************
```

The above files are the main files needed to configure the MAXI board, if more additional example files like the GenericTCPClient or Server files are needed, please add them from the TCPIP Demo folder.

Initializing Analog Inputs

```
// Defining which analog input you want to initialize
// To determine which input is wanted, look in the board specification manual
// AN3 is chosen here (Onboard temperature sensor)

        ADCON0bits.CHS0 = 1;
        ADCON0bits.CHS1 = 1;
        ADCON0bits.CHS2 = 0;
        ADCON0bits.CHS3 = 0;

// AN3 should already be set up as an analog input
        ADCON0bits.GO = 1;

// Wait until A/D conversion is done
        while(ADCON0bits.GO);


// Convert temperature result into ASCII string
        tmp = *((WORD*)(&ADRESL));

// do what you need to do with the value here

// Once done convert 10-bit value into ASCII string
        uitoa(tmp, TEMPString);

// TEMPString is the final output value and should be initialized somewhere within
// your code as a byte. EXAMPLE:

        BYTE TEMPString[8];  //Should always be 8
```

Using LCD

- All subroutine commands for the LCD are located in LCDBlocking.c

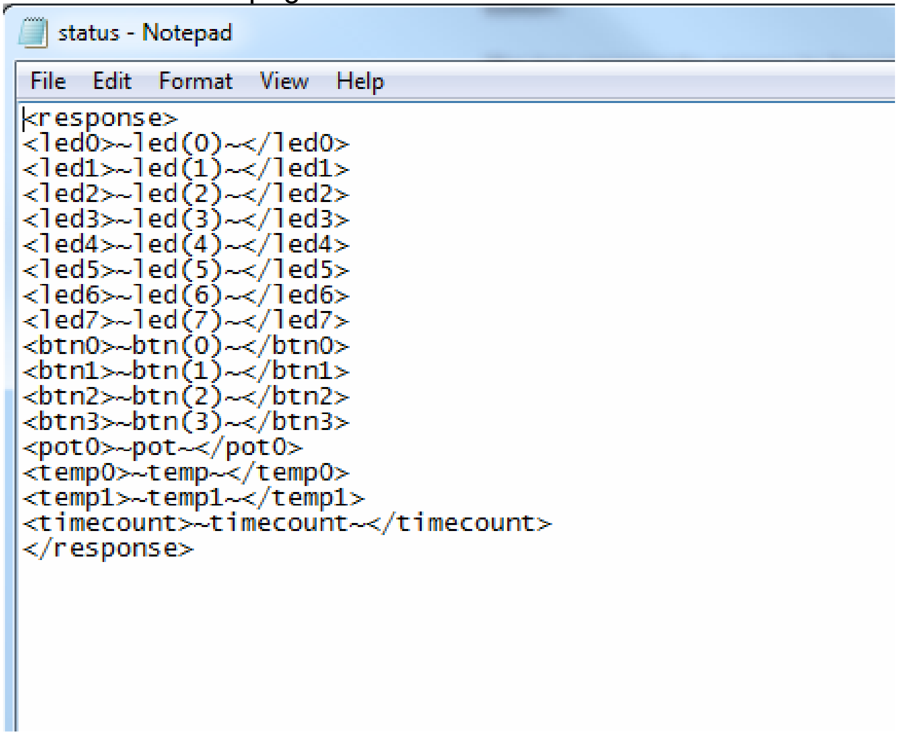
```
// Display character 'T' at position x
// Position goes from Line 1 – 0 →15       & Line 2 - 16 →31
        LCDText [x] = 'T'
```

- after completing all task with the LCD, the subroutine **LCDUpdate();** must be called!!

Modifying Web Server & Homepage

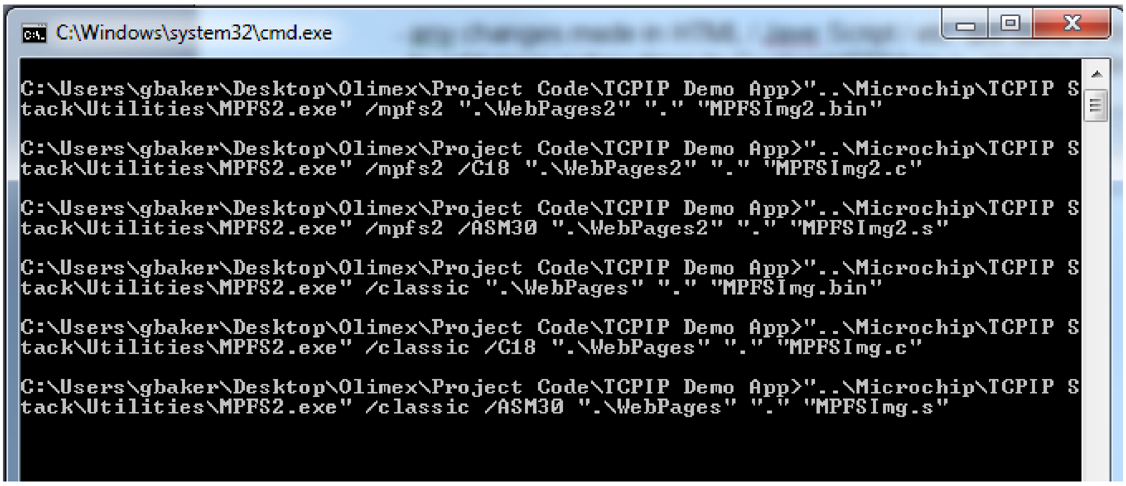
Located in Olimex → Project Code → TCPIP Demo App → Webpages2

- any changes made in HTML / Jave Script / etc. are done in this folder
- to add more subroutines to CustomHTTPApp.c, one must first create a variable in the file → Webpages2 → Status



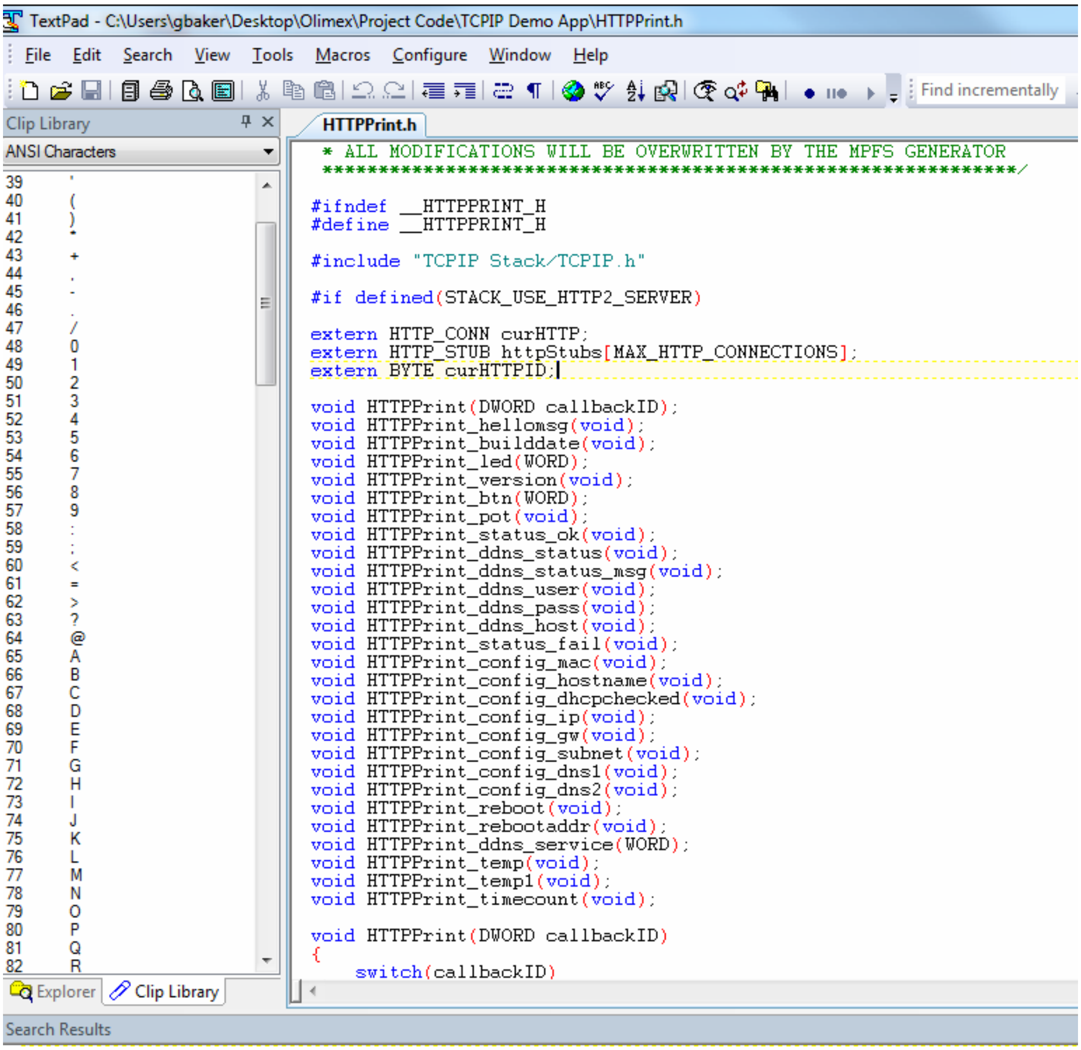
```
<response>
<led0>~led(0)~</led0>
<led1>~led(1)~</led1>
<led2>~led(2)~</led2>
<led3>~led(3)~</led3>
<led4>~led(4)~</led4>
<led5>~led(5)~</led5>
<led6>~led(6)~</led6>
<led7>~led(7)~</led7>
<btn0>~btn(0)~</btn0>
<btn1>~btn(1)~</btn1>
<btn2>~btn(2)~</btn2>
<btn3>~btn(3)~</btn3>
<pot0>~pot~</pot0>
<temp0>~temp~</temp0>
<temp1>~temp1~</temp1>
<timecount>~timecount~</timecount>
</response>
```

- once created return back to the TCPIP Demo App folder and double click on **Convert WebPages to MPFS-Original**, it's a batch file that will execute and create a new HTTPPrint.c file.



```
C:\Users\gbaker\Desktop\Olimex\Project Code\TCPIP Demo App>"..\Microchip\TCPIP S
tack\Utilities\MPFS2.exe" /mpfs2 ".\WebPages2" "." "MPFSImg2.bin"

C:\Users\gbaker\Desktop\Olimex\Project Code\TCPIP Demo App>"..\Microchip\TCPIP S
tack\Utilities\MPFS2.exe" /mpfs2 /C18 ".\WebPages2" "." "MPFSImg2.c"

C:\Users\gbaker\Desktop\Olimex\Project Code\TCPIP Demo App>"..\Microchip\TCPIP S
tack\Utilities\MPFS2.exe" /mpfs2 /ASM30 ".\WebPages2" "." "MPFSImg2.s"

C:\Users\gbaker\Desktop\Olimex\Project Code\TCPIP Demo App>"..\Microchip\TCPIP S
tack\Utilities\MPFS2.exe" /classic ".\WebPages" "." "MPFSImg.bin"

C:\Users\gbaker\Desktop\Olimex\Project Code\TCPIP Demo App>"..\Microchip\TCPIP S
tack\Utilities\MPFS2.exe" /classic /C18 ".\WebPages" "." "MPFSImg.c"

C:\Users\gbaker\Desktop\Olimex\Project Code\TCPIP Demo App>"..\Microchip\TCPIP S
tack\Utilities\MPFS2.exe" /classic /ASM30 ".\WebPages" "." "MPFSImg.s"
```

- if done correctly, one should see the new subroutine define in the HTTPPrint.c file, and then it can be used in the CustomHTTPApp.c



39	'
40	(
41	)
42	*
43	+
44	,
45	-
46	.
47	/
48	0
49	1
50	2
51	3
52	4
53	5
54	6
55	7
56	8
57	9
58	:
59	;
60	<
61	=
62	>
63	?
64	@
65	A
66	B
67	C
68	D
69	E
70	F
71	G
72	H
73	I
74	J
75	K
76	L
77	M
78	N
79	O
80	P
81	Q
82	R

```
 * ALL MODIFICATIONS WILL BE OVERWRITTEN BY THE MPFS GENERATOR
 **************************************************************/

#ifndef __HTTPPRINT_H
#define __HTTPPRINT_H

#include "TCPIP Stack/TCPIP.h"

#if defined(STACK_USE_HTTP2_SERVER)

extern HTTP_CONN curHTTP;
extern HTTP_STUB httpStubs[MAX_HTTP_CONNECTIONS];
extern BYTE curHTTPID;

void HTTPPrint(DWORD callbackID);
void HTTPPrint_hellomsg(void);
void HTTPPrint_builddate(void);
void HTTPPrint_led(WORD);
void HTTPPrint_version(void);
void HTTPPrint_btn(WORD);
void HTTPPrint_pot(void);
void HTTPPrint_status_ok(void);
void HTTPPrint_ddns_status(void);
void HTTPPrint_ddns_status_msg(void);
void HTTPPrint_ddns_user(void);
void HTTPPrint_ddns_pass(void);
void HTTPPrint_ddns_host(void);
void HTTPPrint_status_fail(void);
void HTTPPrint_config_mac(void);
void HTTPPrint_config_hostname(void);
void HTTPPrint_config_dhcpchecked(void);
void HTTPPrint_config_ip(void);
void HTTPPrint_config_gw(void);
void HTTPPrint_config_subnet(void);
void HTTPPrint_config_dns1(void);
void HTTPPrint_config_dns2(void);
void HTTPPrint_reboot(void);
void HTTPPrint_rebootaddr(void);
void HTTPPrint_ddns_service(WORD);
void HTTPPrint_temp(void);
void HTTPPrint_temp1(void);
void HTTPPrint_timecount(void);

void HTTPPrint(DWORD callbackID)
{
    switch(callbackID)
```

- Keep in mind that the new variable must be defined somewhere within the index.html file also in order to be displayed into the webpage, once done double click on **Convert WebPages to MPFS.bat** again before building the project in MPLAB

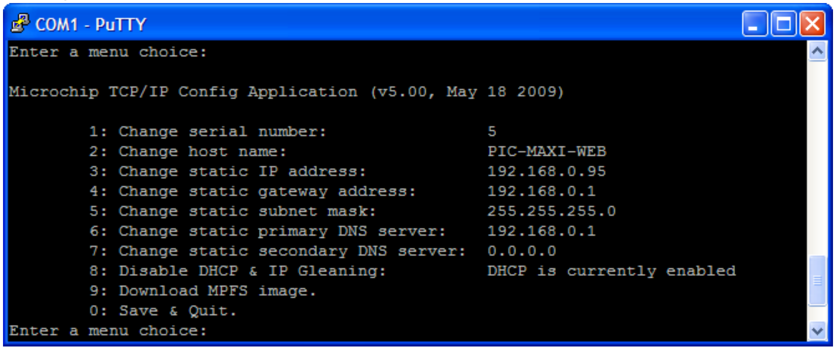
Tera Term (or compatible terminal program: PuTTY, Hyper Terminal)

- settings are go like the following: **19200-8N1**
- used for reading from the serial port of the board
- used for configuring the board

Setup

- Open Tera Term → Select Serial → COM1 → Click OK → Setup → Serial Port.. → Baud Rate = 19200 → Click OK

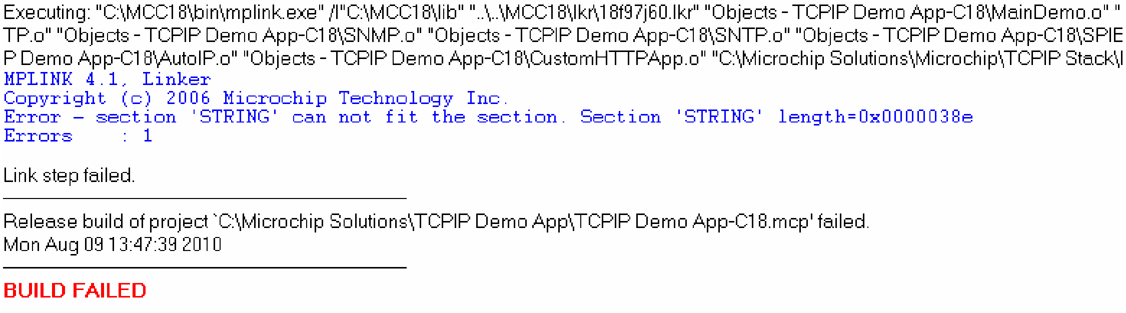
- then run program – the IP address will be printed on the terminal
- in order to enter the configuration menu power on the board with **BUT1** pressed, then release BUT1:



```
Enter a menu choice:

Microchip TCP/IP Config Application (v5.00, May 18 2009)

        1: Change serial number:                5
        2: Change host name:                    PIC-MAXI-WEB
        3: Change static IP address:            192.168.0.95
        4: Change static gateway address:       192.168.0.1
        5: Change static subnet mask:           255.255.255.0
        6: Change static primary DNS server:    192.168.0.1
        7: Change static secondary DNS server:  0.0.0.0
        8: Disable DHCP & IP Gleaning:          DHCP is currently enabled
        9: Download MPFS image.
        0: Save & Quit.
Enter a menu choice:
```

Common Errors

Section Not Fitting Errors

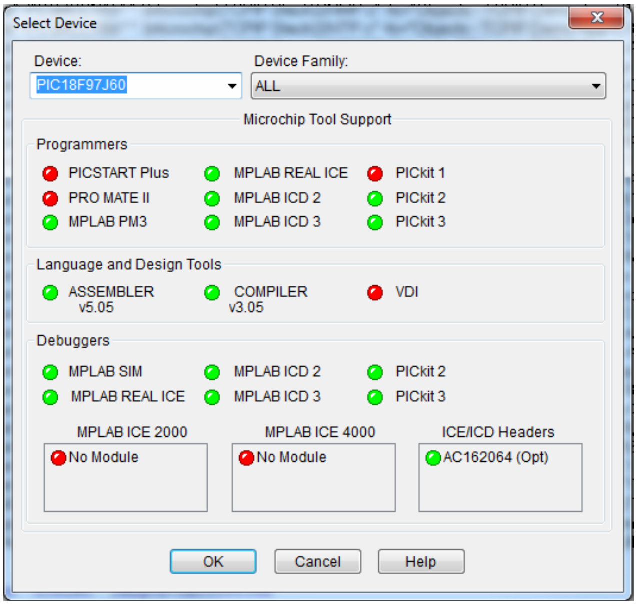


```
Executing: "C:\MCC18\bin\mplink.exe" /l"C:\MCC18\lib" "..\..\MCC18\lkr\18f97j60.lkr" "Objects - TCPIP Demo App-C18\MainDemo.o" "
TP.o" "Objects - TCPIP Demo App-C18\SNMP.o" "Objects - TCPIP Demo App-C18\SNTP.o" "Objects - TCPIP Demo App-C18\SPIE
P Demo App-C18\AutoIP.o" "Objects - TCPIP Demo App-C18\CustomHTTPApp.o" "C:\Microchip Solutions\Microchip\TCPIP Stack\I
MPLINK 4.1, Linker
Copyright (c) 2006 Microchip Technology Inc.
Error - section 'STRING' can not fit the section. Section 'STRING' length=0x0000038e
Errors    : 1

Link step failed.
----------------------------------------------------------------------
Release build of project `C:\Microchip Solutions\TCPIP Demo App\TCPIP Demo App-C18.mcp' failed.
Mon Aug 09 13:47:39 2010
----------------------------------------------------------------------
BUILD FAILED
```

- If an error similar to above that is saying some certain file cannot fit when building the project, it means that code for the project is too big to be stored onto the device and needs to be shorter.

Note: *In some situations this error can be solved by editing the linker script. MCC18 will only link correctly if a file requires less RAM than the size of a RAM page as defined in the linker script. So if you increase the size of a particular* *section you may be able to link successfully. This, however, is not true for FLASH memory.*

- Also make sure that the proper device is selected, under Configure → Select Devices → PIC18F97J60



Device: PIC18F97J60

Device Family: ALL

Microchip Tool Support

Programmers

🔴 PICSTART Plus 🟢 MPLAB REAL ICE 🔴 PICkit 1
🔴 PRO MATE II 🟢 MPLAB ICD 2 🟢 PICkit 2
🟢 MPLAB PM3 🟢 MPLAB ICD 3 🟢 PICkit 3

Language and Design Tools

🟢 ASSEMBLER v5.05 🟢 COMPILER v3.05 🔴 VDI

Debuggers

🟢 MPLAB SIM 🟢 MPLAB ICD 2 🟢 PICkit 2
🟢 MPLAB REAL ICE 🟢 MPLAB ICD 3 🟢 PICkit 3

MPLAB ICE 2000: 🔴 No Module

MPLAB ICE 4000: 🔴 No Module

ICE/ICD Headers: 🟢 AC162064 (Opt)

Global Variable Errors (not sure what this is for, hope it is helpful, however)
- If one tries to program a global variable in any particular way they might run into some problems as this is not normal C language so certain methods are need to do that

One method would be:

```
extern int GlobalVariable
```

- *NOTE* this method will not always work, however it has been used in the project codes before

Syntax Errors
- this is a very common error, when double clicking the error, it brings you to the location that is causing the problem, however the pointer sometimes point below the line with the actual error. MCC18 is famous for its ambiguous explanatory messages so good luck.

Many thanks to Tom Nguyen for the great help with this manual :)