



**DesignNews**

## PIC Microcontroller Embedded Development Using the CCS PIC MCU C Compiler

**Day 2:**

## TCP/IP Development Using the CCS C Compiler

Sponsored by



## Webinar Logistics

- Turn on your system sound to hear the streaming presentation.
- If you have technical problems, click “Help” or submit a question asking for assistance.
- Participate in ‘Attendee Chat’ by maximizing the chat widget in your dock.



## Fred Eady

Visit 'Lecturer Profile' in your console for more details.

# AGENDA

- **Create a PIC18LF4620 TCP/IP Project**
- **Write the Application Code**
- **Compile and Run the Application Code**
- **Telnet Server App**

ABCDBUGS

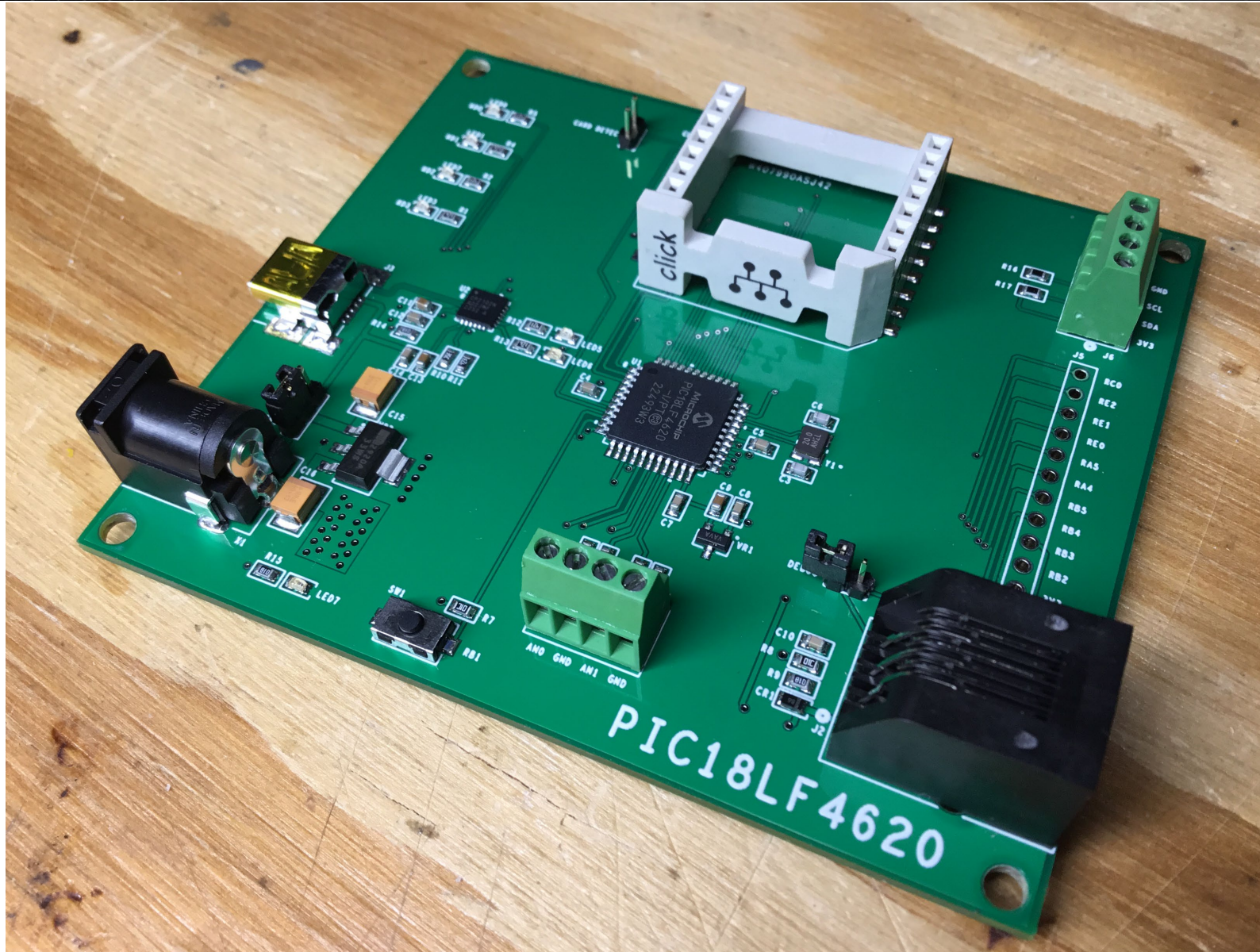
LMRNOBUGS

OSARBUGSCDEDBDIS

LILBMRBUGS



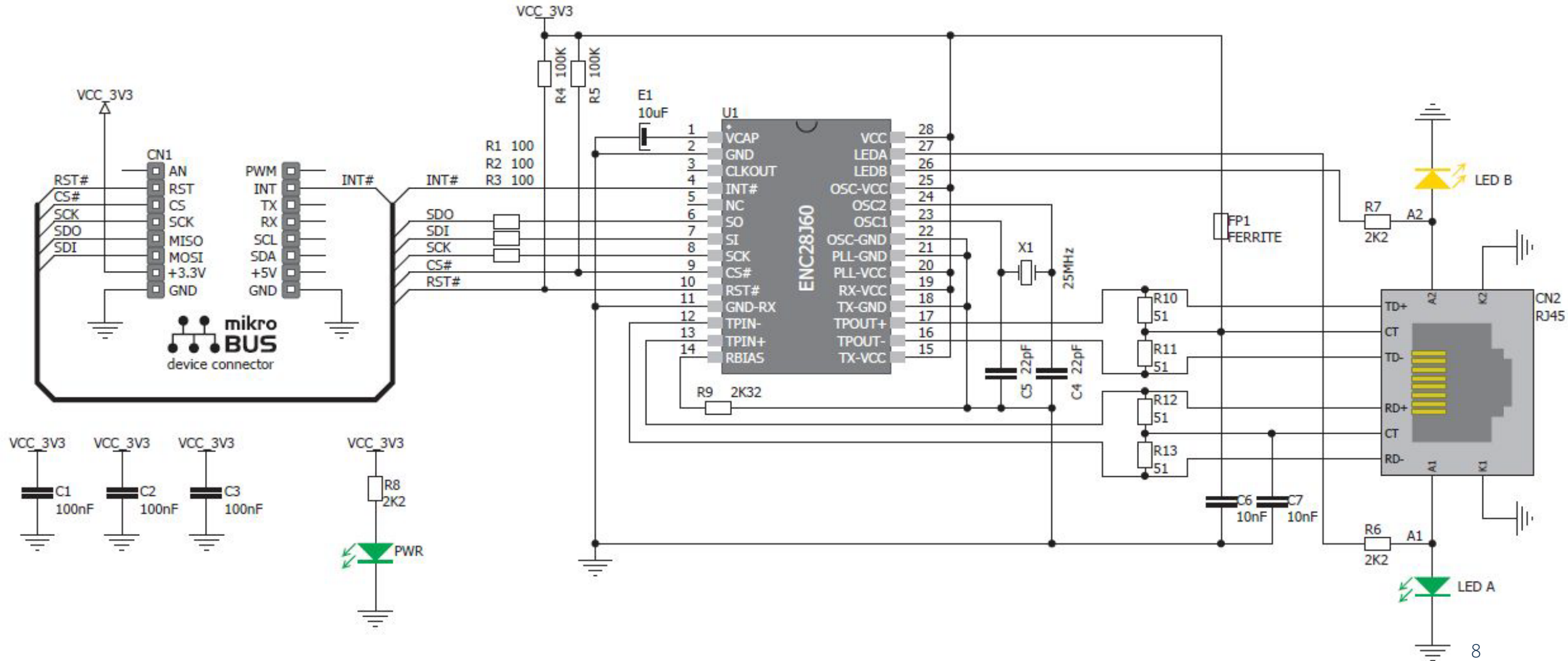
## PIC18LF4620 Hardware







# ENC28J60 *click* Hardware





## Choose the PIC and Specify the CPU Clock Speed

Project Wizard - C:\Users\Public\cecCCS\day2\_code\main.ccsproj

File Help

Peripherals

- Analog
- Communications
- SPI
- Drivers
- Header Files
- High/Low Voltage
- Interrupts
- I/O Pins
- Timers 0-2
- Timers 3-up
- LCD (Internal)
- LCD (External)
- Capacitive Touch
- RTCC
- CCP/ref

Options Code

General

Device

Family: PIC18  Debug Code

Device: PIC18LF4620 Fixed Compiler Version: None

Clock

Oscillator Type: Crystal  Use USB Low Speed

Crystal Clock Speed: 20 MHz  Use USB Full Speed

CPU Clock Speed: 20 MHz 5 MIPS  Clock Out

WDT

Enabled

Check any of the following to restart WDT during calls to:

getc() and fgetc()

i2c\_read()

Delay Functions

Reset: 4.0 ms

WDT Reset

- 4 ms
- 8 ms
- 16 ms
- 32 ms
- 64 ms
- 128 ms
- 256 ms
- 512 ms
- 1024 ms
- 2048 ms
- 4096 ms
- 8192 ms
- 16384 ms
- 32768 ms
- 65536 ms
- 131072 ms

Chip: PIC18LF4620 Frequency: 20,000,000

Create Project Cancel

## Define the Analog Pins

Project Wizard - C:\Users\Public\cecCCS\day2\_code\main.ccsproj

File Help

Peripherals

- Analog
- Communications
- SPI
- Drivers
- Header Files
- High/Low Voltage
- Interrupts
- I/O Pins
- Timers 0-2
- Timers 3-up
- LCD (Internal)
- LCD (External)
- Capacitive Touch
- RTCC
- CCP/ref

Options Code

Analog Input

Analog Pins

- None
- A0 A1 A2 A3 A5 E0 E1 E2 B2 B3 B1 B4 B0
- A0 A1 A2 A3 A5 E0 E1 E2 B2 B3 B1 B4
- A0 A1 A2 A3 A5 E0 E1 E2 B2 B3 B1
- A0 A1 A2 A3 A5 E0 E1 E2 B2 B3
- A0 A1 A2 A3 A5 E0 E1 E2 B2
- A0 A1 A2 A3 A5 E0 E1 E2
- A0 A1 A2 A3 A5 E0 E1
- A0 A1 A2 A3 A5 E0
- A0 A1 A2 A3 A5
- A0 A1 A2 A3
- A0 A1 A2
- A0 A1
- A0

Range VrefL-VrefH

Units: 0-1023

Internal RC Clock

Acquisition time: 1.6 us

Chip: PIC18LF4620 Frequency: 20,000,000

Create Project Cancel

## Configure the UART

Project Wizard - C:\Users\Public\cecCCS\day2\_code\main.ccsproj

File Help

Peripherals

- Analog
- Communications
- SPI
- Drivers
- Header Files
- High/Low Voltage
- Interrupts
- I/O Pins
- Timers 0-2
- Timers 3-up
- LCD (Internal)
- LCD (External)
- Capacitive Touch
- RTCC
- CCP/ref

Options Code

Communications

RS-232

Use RS-232

Port Count:  1  2  3  4

Type:  Standard  RS232  RS485

Restart WDT on RS232

Baud:  Parity:   Invert  Float\_high  Errors  External interrupt

Transmit Pin:  Receive Pin:   Enable Pin:  Receive Enable Pin:

Bits:  Stream:  Buffer Size:  ID:

I2C

Use I2C

SDA:  SCL:   Master  Slave  Restart WDT on I2C  Force Hardware

Fast  Slow Slave Address:

Chip: PIC18LF4620 Frequency: 20,000,000 Create Project Cancel

## Define the SPI Portal

Project Wizard - C:\Users\Public\cecCCS\day2\_code\main.ccsproj

File Help

Options Code

Peripherals

- Analog
- Communications
- SPI**
- Drivers
- Header Files
- High/Low Voltage
- Interrupts
- I/O Pins
- Timers 0-2
- Timers 3-up
- LCD (Internal)
- LCD (External)
- Capacitive Touch
- RTCC
- CCP/ref

SPI

Port Count: 1 PORT 1

Pin Assignments

Use hardware

SPI1

CLK pin: C3

DO pin: C5

DI pin: C4

Enable Pin: C2

D diagnostic Pin: None

Load Pin: None

Timing

Baud: [dropdown]

Baud: 0

High time(us): 0

Low time(us): 0

Enable delay(ms): 0

Data hold(ms): 0

Settings

Master

Mode: 0

Bits: 8

Load active: Low

Enable active: Low

First bit: MSB

Sample Count: 1

Stream: [text box]

C2>

C3>

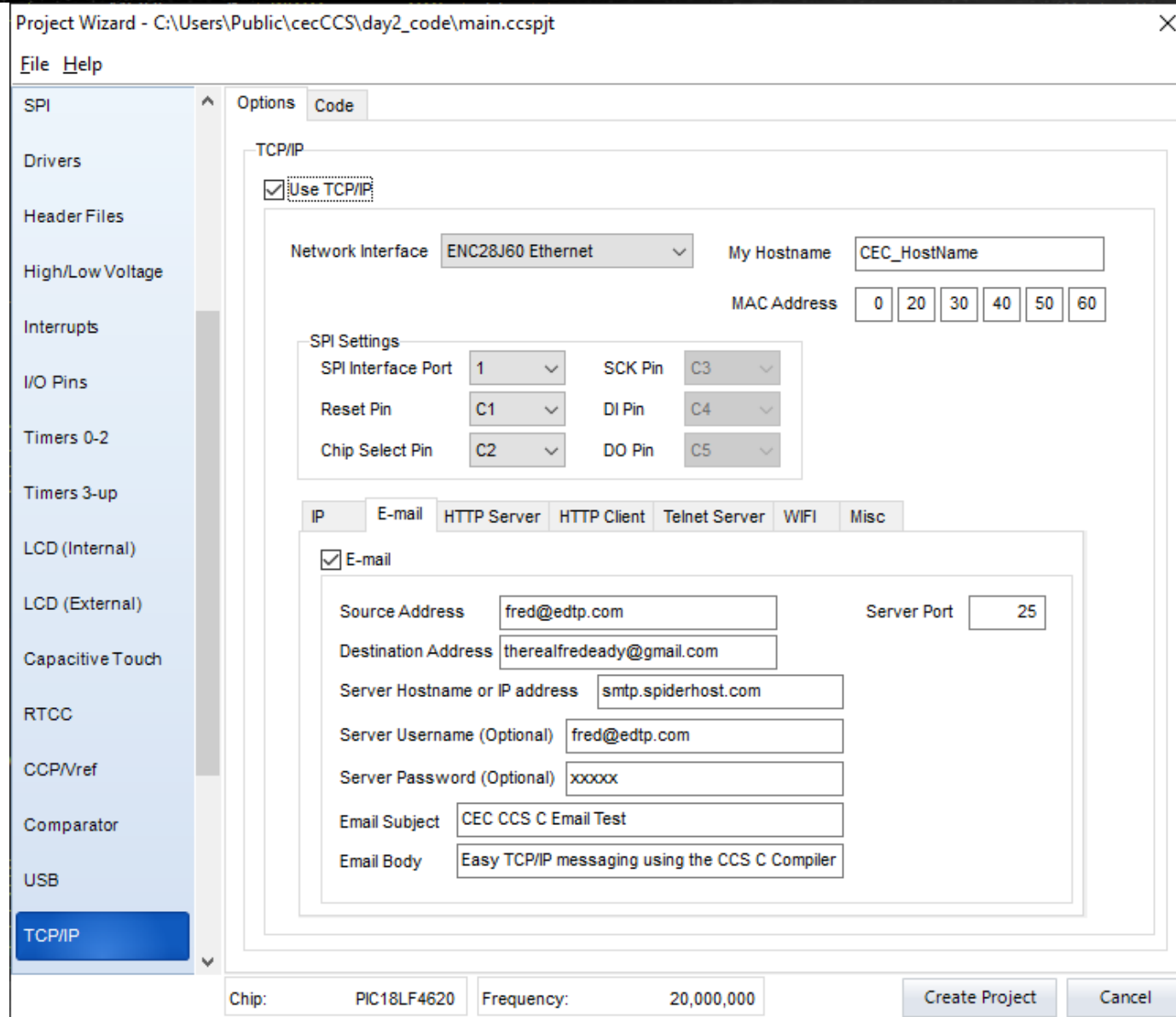
C4< D7 D6 D5 D4 D3 D2 D1 D0

C5> D7 D6 D5 D4 D3 D2 D1 D0

Chip: PIC18LF4620 Frequency: 20,000,000

Create Project Cancel

## Define the Email Variables



The screenshot shows the Project Wizard interface for a PIC18LF4620 project. The 'Options' tab is active, and the 'Code' sub-tab is selected. The 'TCP/IP' section is expanded, showing the following settings:

- Use TCP/IP
- Network Interface: ENC28J60 Ethernet
- My Hostname: CEC\_HostName
- MAC Address: 0 20 30 40 50 60
- SPI Settings:
  - SPI Interface Port: 1
  - SCK Pin: C3
  - Reset Pin: C1
  - DI Pin: C4
  - Chip Select Pin: C2
  - DO Pin: C5

The 'E-mail' sub-tab is also active, showing the following settings:

- E-mail
- Source Address: fred@edtp.com
- Server Port: 25
- Destination Address: therealfreedy@gmail.com
- Server Hostname or IP address: smtp.spiderhost.com
- Server Username (Optional): fred@edtp.com
- Server Password (Optional): xxxxxx
- Email Subject: CEC CCS C Email Test
- Email Body: Easy TCP/IP messaging using the CCS C Compiler

At the bottom of the window, the 'Chip' is set to PIC18LF4620 and the 'Frequency' is 20,000,000. 'Create Project' and 'Cancel' buttons are visible.

## Set and Expose the Fuses and Create the Project

Project Wizard - C:\Users\Public\cecCCS\day2\_code\main.ccsjpt

File Help

Timers 0-2

Timers 3-up

LCD (Internal)

LCD (External)

Capacitive Touch

RTCC

CCP/Vref

Comparator

USB

TCP/IP

MODBus

CANBus

Bootloader

Advanced

Options

Fuses

Options Code

Fuses

Include Fuses

Fail-safe clock monitor enabled

Internal External Switch Over mode enabled

Power Up Timer

No brownout reset

Brownout reset at 2.1V

PORTB pins are configured as analog input channels on RESET

Timer1 configured for low-power operation

Master Clear pin enabled

Stack full/underflow will cause reset

Low Voltage Programming on B3(PIC16) or B5(PIC18)

Extended set extension and Indexed Addressing mode enabled

Code protected from reads

Boot Block Code Protected

Data EEPROM Code Protected

Program Memory Write Protected

Configuration registers write protected

Boot block write protected

Data EEPROM write protected

Memory protected from table reads

Boot block protected from table reads

Chip: PIC18LF4620 Frequency: 20,000,000

Create Project Cancel

## main.h

```

#include <18LF4620.h>
#define ADC=10

/*
TCP/IP Stack enabled.
Many TCP/IP configuration settings (servers enabled, ports used,
etc) are defined in TCPIPConfig.h.
Many hardware configuration settings (SPI port and GPIO pins used)
are defined in HardwareProfile.h.
*/

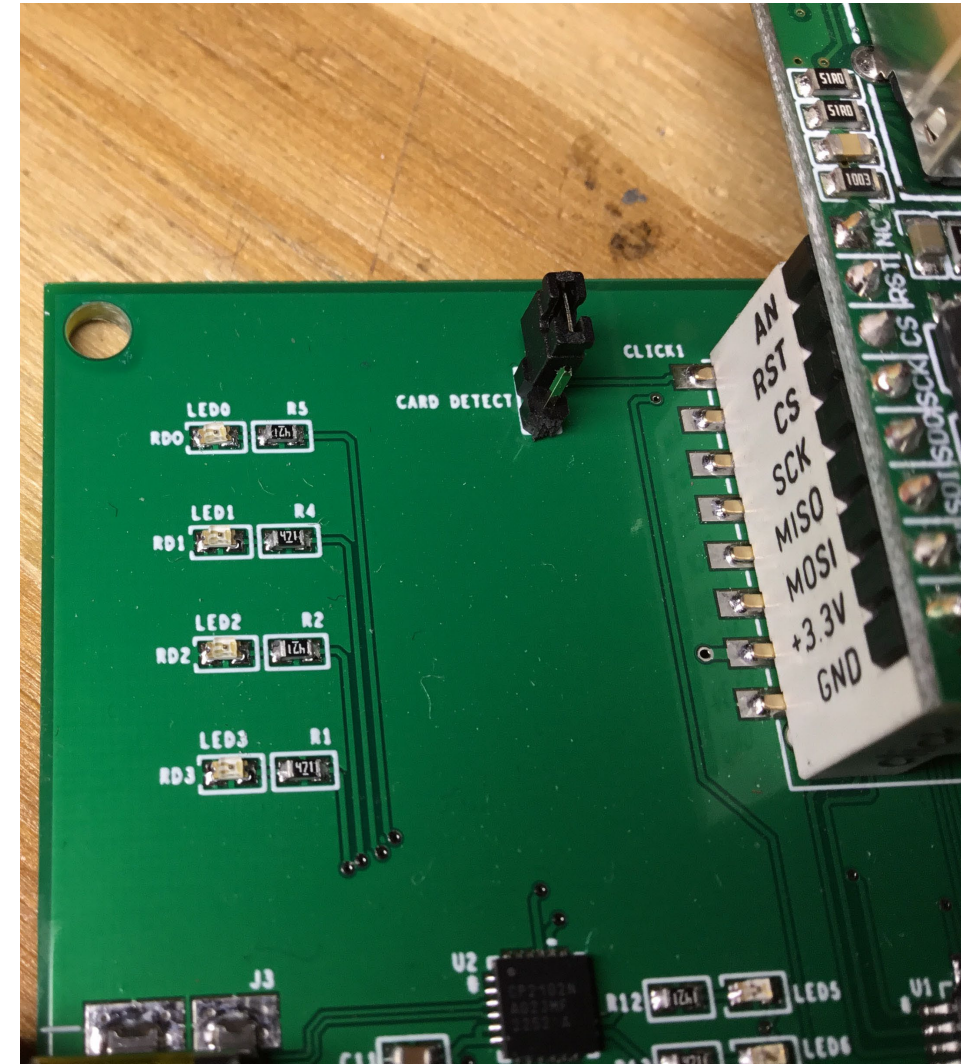
#include "tcpip/p18cxxx.h"

#FUSES NOWDT           //No Watch Dog Timer
#FUSES NOBROWNOUT     //No brownout reset
#FUSES BORV21         //Brownout reset at 2.1V
#FUSES PBADEN         //PORTB pins are configured as analog input channels on RESET
#FUSES LPT10SC        //Timer1 configured for low-power operation
#FUSES MCLR           //Master Clear pin enabled
#FUSES NOPROTECT      //Code not protected from reading
#FUSES NOWRT          //Program memory not write protected
#FUSES NOEBTR         //Memory not protected from table reads

#use delay(crystal=20MHz)
#use FIXED_IO( D_outputs=PIN_D3,PIN_D2,PIN_D1,PIN_D0 )
#define btnB1          PIN_B1
#define led0_ORG_D0    PIN_D0
#define led1_BLU_D1    PIN_D1
#define led2_RED_D2    PIN_D2
#define led3_GRN_D3    PIN_D3

#use rs232(baud=115200,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8,stream=PORT1)
#use i2c(Master,Fast,sda=PIN_D4,scl=PIN_D5)

```



## main.h

```

#define MIN(a,b) ((a > b) ? b : a)

#include <stdint.h>
#include "tcpip/StackTsk2.h"
#include "tcpip/TCPIPConfig.h"
#include "tcpip/HardwareProfile.h"

typedef struct
{
    BYTE vSocketPurpose;
    BYTE vMemoryMedium;
    WORD wTXBufferSize;
    WORD wRXBufferSize;
} TCPSocketInitializer_t;

#if TCP_CONFIGURATION > 0
TCPSocketInitializer_t TCPSocketInitializer[TCP_CONFIGURATION] =
{
    #if defined(STACK_USE_CCS_HTTP2_SERVER)
    {TCP_PURPOSE_HTTP_SERVER, TCP_ETH_RAM, STACK_CCS_HTTP2_SERVER_TX_SIZE, STACK_CCS_HTTP2_SERVER_RX_SIZE},
    #endif
    #if defined(STACK_USE_SMTP_CLIENT)
    {TCP_PURPOSE_DEFAULT, TCP_ETH_RAM, STACK_CCS_SMTP_TX_SIZE, STACK_CCS_SMTP_RX_SIZE},
    #endif
    #if defined(STACK_USE_MY_TELNET_SERVER)
    {TCP_PURPOSE_TELNET, TCP_ETH_RAM, STACK_MY_TELNET_SERVER_TX_SIZE, STACK_MY_TELNET_SERVER_RX_SIZE},
    #endif
    #if defined(STACK_USE_CCS_HTTP_CLIENT)
    {TCP_PURPOSE_GENERIC_TCP_CLIENT, TCP_ETH_RAM, STACK_MY_HTTPC_TX_SIZE, STACK_MY_HTTPC_RX_SIZE},
    #endif
};
#else
#undef TCP_CONFIGURATION
#define TCP_CONFIGURATION 1
TCPSocketInitializer_t TCPSocketInitializer[TCP_CONFIGURATION] =
{
    {TCP_PURPOSE_DEFAULT, TCP_ETH_RAM, 250, 250}
};
#endif

#include "tcpip/StackTsk2.c"

```

IP	E-mail	HTTP Server	HTTP Client	Telnet Server	WIFI	Misc
<input checked="" type="checkbox"/> E-mail						
Source Address	<input type="text" value="fred@edtp.com"/>					Server
Destination Address	<input type="text" value="therealfreedy@gmail.com"/>					
Server Hostname or IP address	<input type="text" value="smtp.spiderhost.com"/>					
Server Username (Optional)	<input type="text" value="fred@edtp.com"/>					
Server Password (Optional)	<input type="text" value="xxxxx"/>					
Email Subject	<input type="text" value="CEC CCS C Email Test"/>					
Email Body	<input type="text" value="Easy TCP/IP messaging using the CCS C Compiler"/>					



## TCPIPconfig.h

```

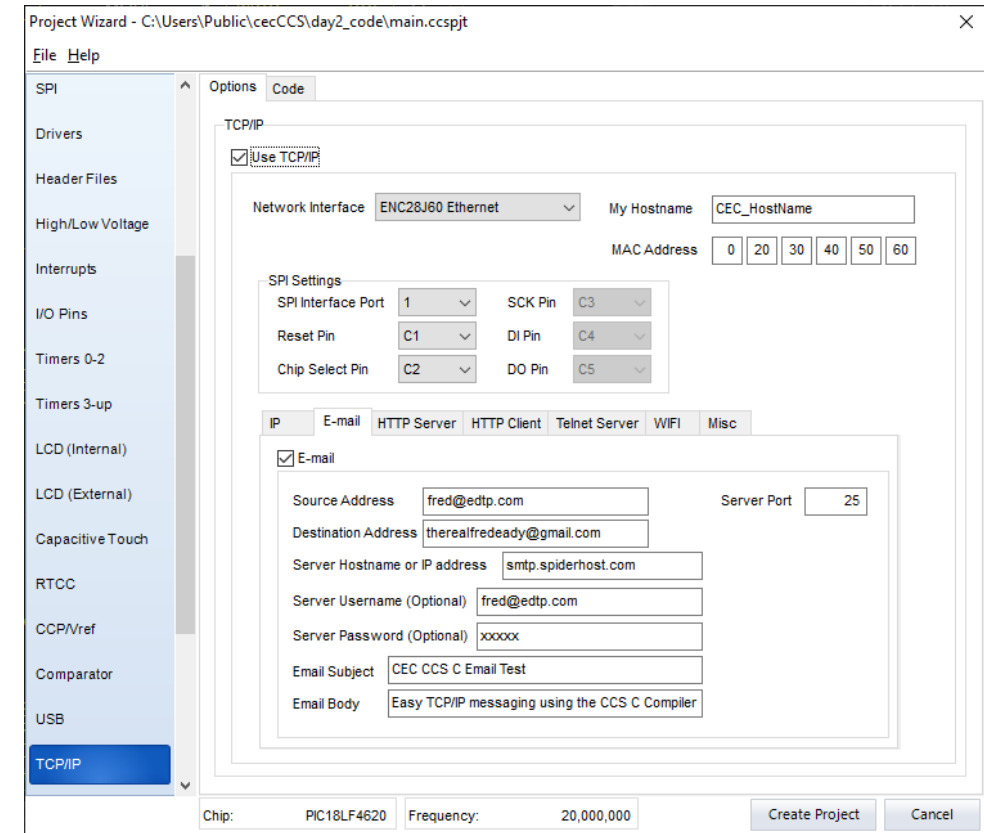
#ifndef __TCPIP_CONFIG_H
#define __TCPIP_CONFIG_H

#include "GenericTypeDefs.h"
#include "Compiler.h"

#define STACK_USE_ICMP_SERVER 1
#define STACK_USE_TCP 1
#define STACK_USE_DHCP_CLIENT 1

//Email Options
#define STACK_USE_DNS
#define STACK_USE_SMTP_CLIENT
#define STACK_USE_SMTP
#define STACK_CCS_SMTP_TX_SIZE 250
#define STACK_CCS_SMTP_RX_SIZE 250
#define EMAIL_PORT (25u)
#define EMAIL_SERVER "smtp.spiderhost.com"
#define EMAIL_TO "therealfreedy@gmail.com"
#define EMAIL_FROM "fred@edtp.com"
#define EMAIL_SUBJECT "CEC CCS C Email Test"
#define EMAIL_BODY "Easy TCP/IP messaging using the CCS C Compiler\r\n.\r\n"
#define EMAIL_CC ""
#define EMAIL_BCC ""
#define EMAIL_OTHERHEADERS ""
#define SMTP_AUTH 1
#define EMAIL_USERNAME "fred@edtp.com"
#define EMAIL_PASSWORD "xxxxx"

```



## TCPIPconfig.h

```
#define MY_DEFAULT_HOST_NAME
```

"CEC\_HostName"

```
#define MY_DEFAULT_MAC_BYTE1
```

(0x0)

```
#define MY_DEFAULT_MAC_BYTE2
```

(0x20)

```
#define MY_DEFAULT_MAC_BYTE3
```

(0x30)

```
#define MY_DEFAULT_MAC_BYTE4
```

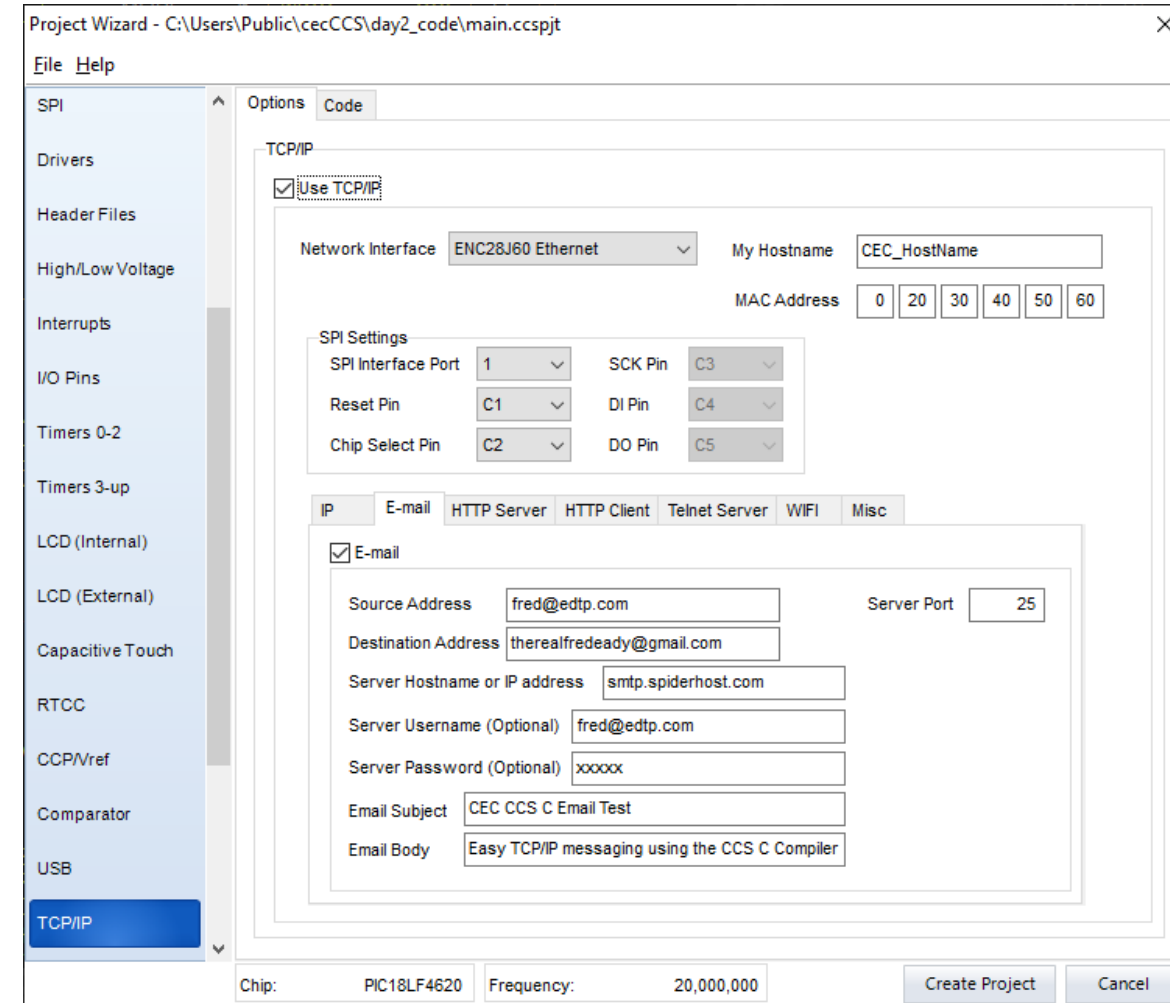
(0x40)

```
#define MY_DEFAULT_MAC_BYTE5
```

(0x50)

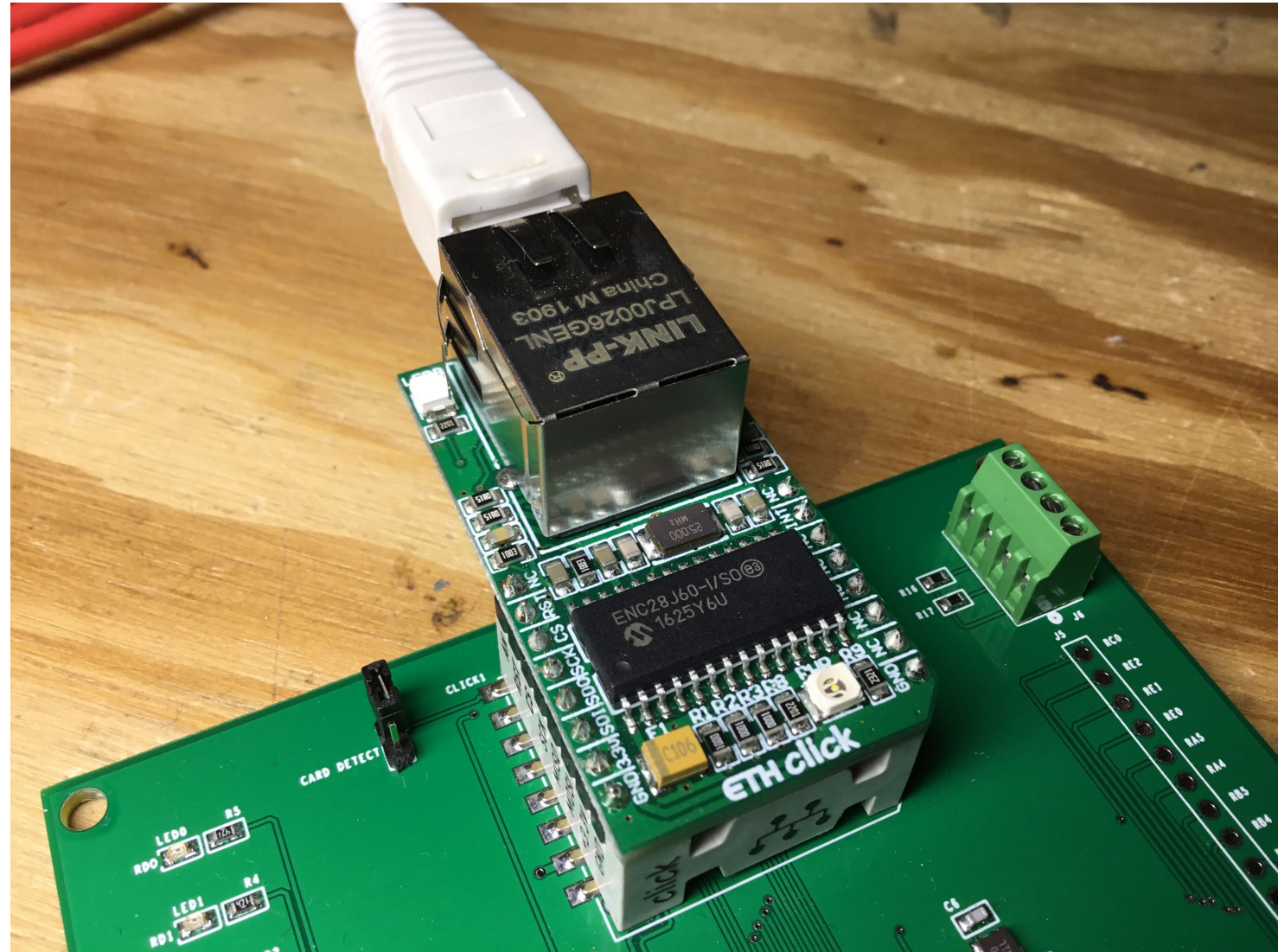
```
#define MY_DEFAULT_MAC_BYTE6
```

(0x60)



## Send Email Code

```
void EmailNow(void)
{
    if (!SMTPBeginUsage())
    {
        SMTPEndUsage();
        SMTPBeginUsage();
    }
    EmailInit();
    SMTPSendMail();
}
```



## Main Send Email Application Code

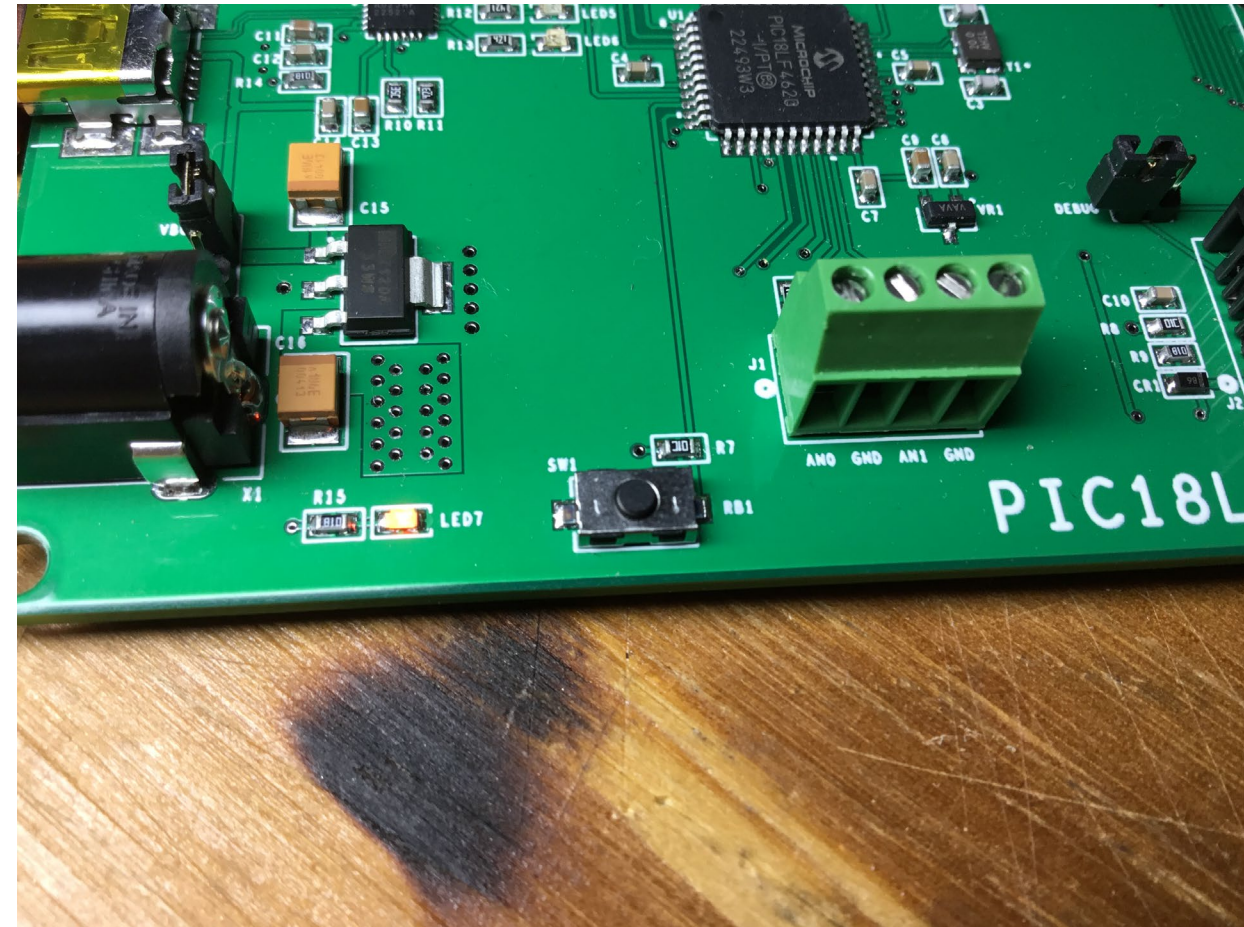
```
void main()
{
    setup_adc_ports(AN0_TO_AN1, VREF_VREF);
    setup_adc(ADC_CLOCK_INTERNAL | ADC_TAD_MUL_8);

    IPAddressInit();
    EmailInit();
    TickInit();
    enable_interrupts(GLOBAL);
    StackInit();

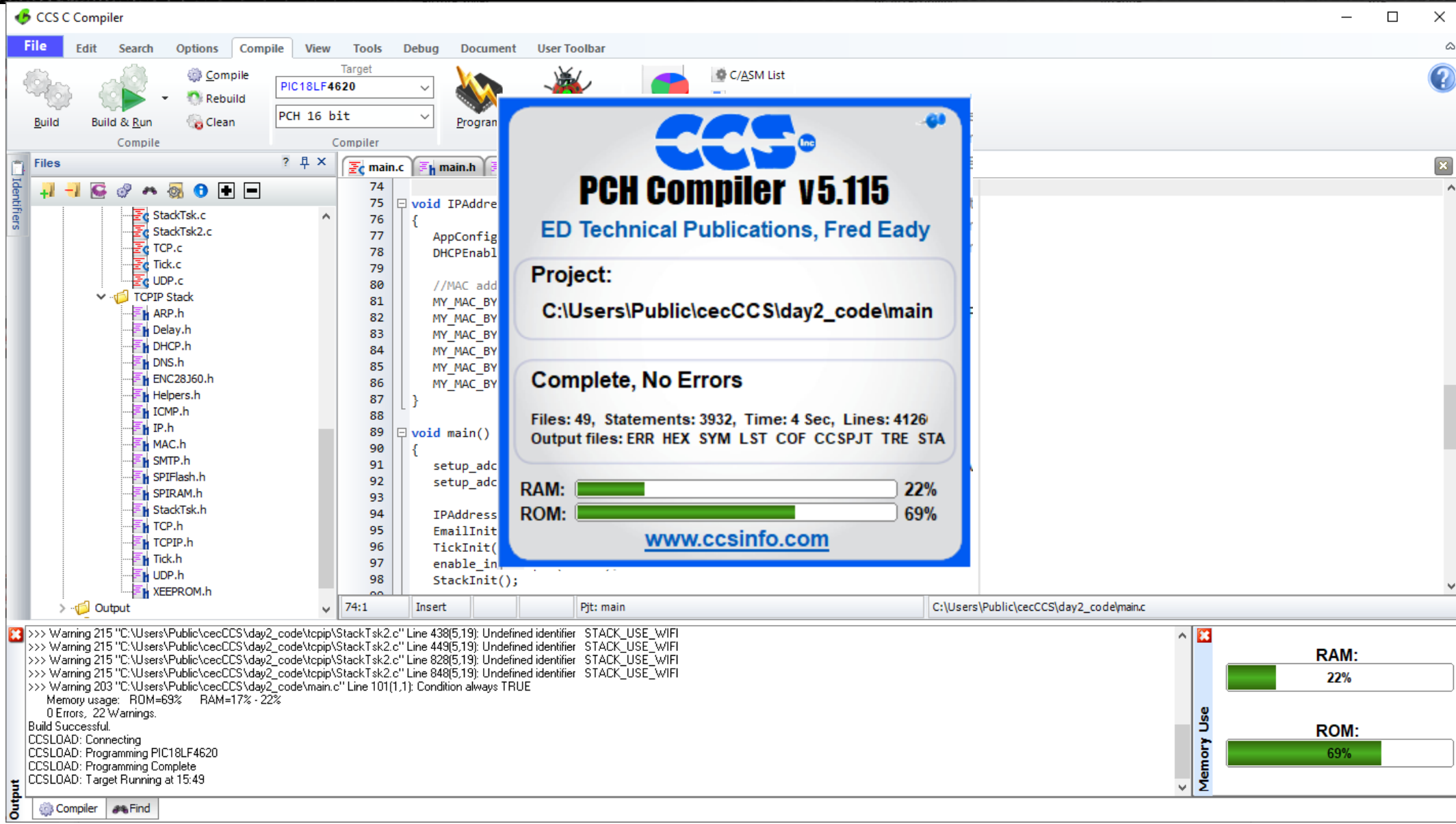
    while(TRUE)
    {
        // TCP/IP code
        StackTask();

        StackApplications();

        /*TODO: Add logic for calling EmailNow(). The EmailNow() function
        starts the SMTP state machine and sends the email. EmailNow() can
        be called, for example, on a pushbutton press or a timer event. */
        if(!input(btnB1))
        {
            delay_ms(10);
            if(!input(btnB1))
            {
                EmailNow();
                while(!input(btnB1));
                delay_ms(2000);
            }
        }
    }
}
```



# Compile the Application Code



**CCS C Compiler**

File Edit Search Options Compile View Tools Debug Document User Toolbar

Build Build & Run Clean

Target: PIC18LF4620  
PCH 16 bit

Compiler

Files

- StackTsk.c
- StackTsk2.c
- TCP.c
- Tick.c
- UDP.c
- TCPIP Stack
  - ARP.h
  - Delay.h
  - DHCP.h
  - DNS.h
  - ENC28J60.h
  - Helpers.h
  - ICMP.h
  - IP.h
  - MAC.h
  - SMTP.h
  - SPIFlash.h
  - SPIRAM.h
  - StackTsk.h
  - TCP.h
  - TCPIP.h
  - Tick.h
  - UDP.h
  - XEEPROM.h

```
74
75 void IPAddr
76 {
77     AppConfig
78     DHCPEnabl
79
80     //MAC add
81     MY_MAC_BY
82     MY_MAC_BY
83     MY_MAC_BY
84     MY_MAC_BY
85     MY_MAC_BY
86     MY_MAC_BY
87 }
88
89 void main()
90 {
91     setup_adc
92     setup_adc
93
94     IPAddress
95     EmailInit
96     TickInit(
97     enable_in
98     StackInit();
99
```

**CCS Inc**  
**PCH Compiler v5.115**  
ED Technical Publications, Fred Eady

**Project:**  
C:\Users\Public\cecCCS\day2\_code\main

**Complete, No Errors**

Files: 49, Statements: 3932, Time: 4 Sec, Lines: 4126  
Output files: ERR HEX SYM LST COF CCSPJT TRE STA

RAM: 22%  
ROM: 69%

[www.ccsinfo.com](http://www.ccsinfo.com)

Output

```
>>> Warning 215 "C:\Users\Public\cecCCS\day2_code\tcpip\StackTsk2.c" Line 438(5,19): Undefined identifier STACK_USE_WIFI
>>> Warning 215 "C:\Users\Public\cecCCS\day2_code\tcpip\StackTsk2.c" Line 449(5,19): Undefined identifier STACK_USE_WIFI
>>> Warning 215 "C:\Users\Public\cecCCS\day2_code\tcpip\StackTsk2.c" Line 828(5,19): Undefined identifier STACK_USE_WIFI
>>> Warning 215 "C:\Users\Public\cecCCS\day2_code\tcpip\StackTsk2.c" Line 848(5,19): Undefined identifier STACK_USE_WIFI
>>> Warning 203 "C:\Users\Public\cecCCS\day2_code\main.c" Line 101(1,1): Condition always TRUE
Memory usage: ROM=69% RAM=17% - 22%
0 Errors, 22 Warnings.
Build Successful.
CCSLOAD: Connecting
CCSLOAD: Programming PIC18LF4620
CCSLOAD: Programming Complete
CCSLOAD: Target Running at 15.49
```

Memory Use

RAM: 22%

ROM: 69%

## Run the Application Code

The screenshot shows a Gmail interface. On the left is a navigation sidebar with a hamburger menu icon, the Gmail logo, and a 'Compose' button. Below are folders: 'Inbox' (207), 'Starred', 'Snoozed', 'Sent', 'Drafts', and 'More'. At the bottom of the sidebar is a 'Labels' section with a plus sign. The main area has a search bar at the top. Below it is a toolbar with icons for back, add, warning, delete, envelope, clock, checkmark, folder, and a menu icon. The email title is 'CEC CCS C Email Test' with an 'Inbox' label. The sender is 'fred@edtp.com' with a circular profile picture containing the letter 'F'. The email body text is 'Easy TCP/IP messaging using the CCS C Compiler'. At the bottom are 'Reply' and 'Forward' buttons.

## main.c

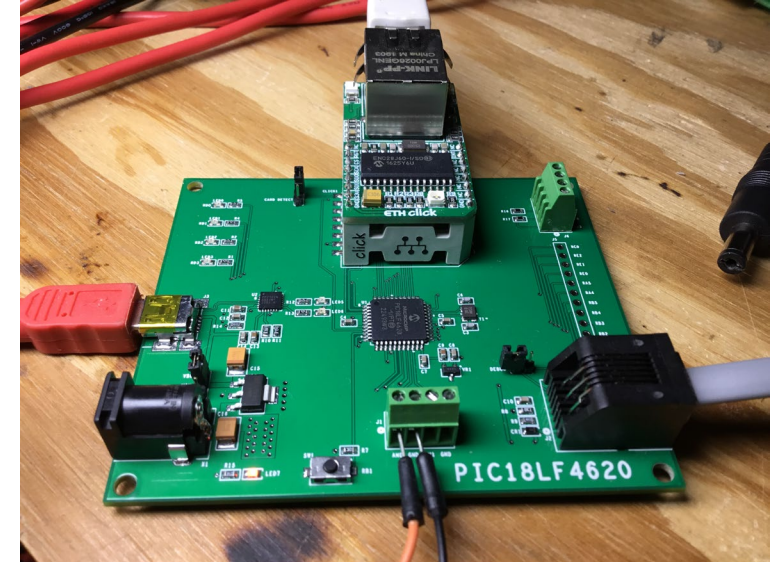
```
void main()
{
    setup_adc_ports(AN0_TO_AN1, VREF_VREF);
    setup_adc(ADC_CLOCK_INTERNAL | ADC_TAD_MUL_8);
    set_adc_channel(0);

    IPAddressInit();
    TickInit();
    enable_interrupts(GLOBAL);
    StackInit();

    g_MyTelnetSock = TCPOpen(0, TCP_OPEN_SERVER, TELNET_PORT, TCP_PURPOSE_TELNET);

    while(TRUE)
    {

        // TCP/IP code
        StackTask();
        StackApplications();
        MyTelnetTask();
    }
}
```



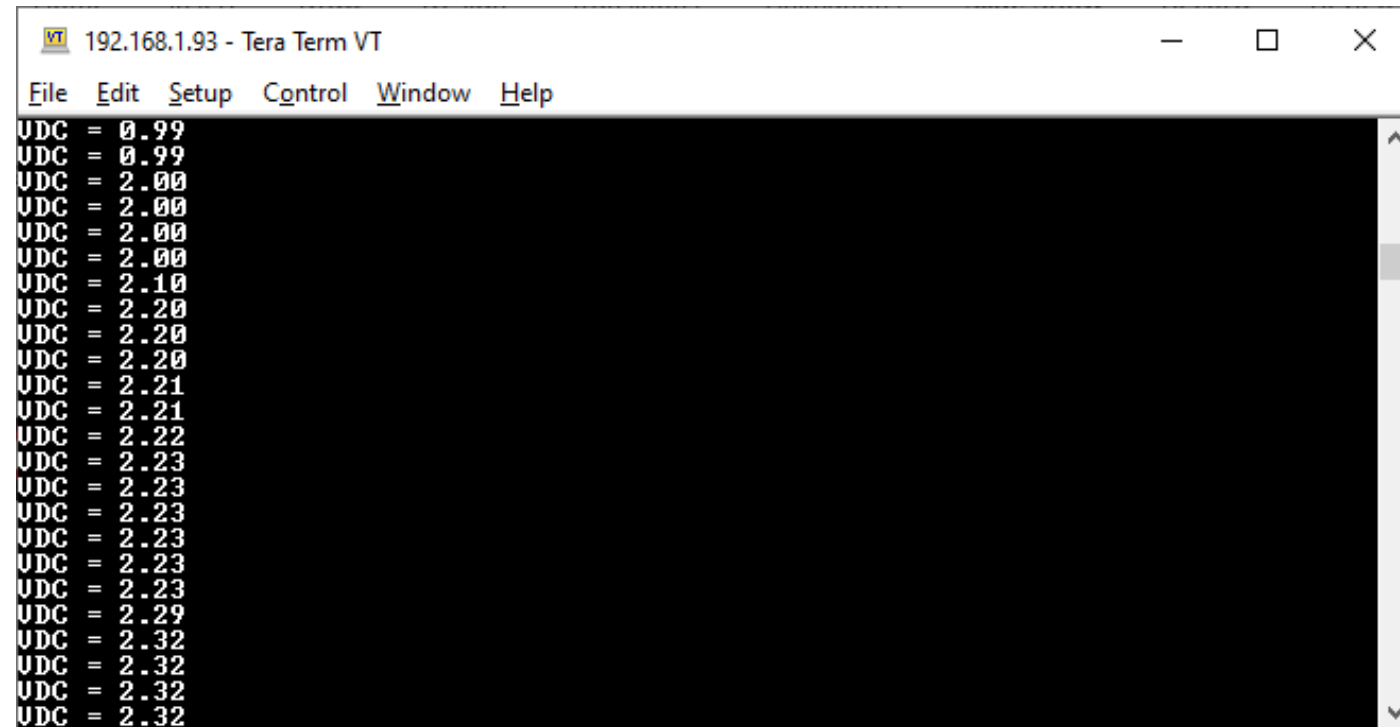
## Telnet Task

```
#include <main.h>

TCP_SOCKET g_MyTelnetSock;

long adcVal;
float conVal = 0.002443793;
unsigned int8 potVal[16];
unsigned int8 bufSpace;

void MyTelnetTask(void)
{
    if (!TCPIsConnected(g_MyTelnetSock))
        return;
    /* TODO: use TCPIsGetReady(g_MyTelnetSock) to determine
    if we have received any data, and if we have then use
    TCPGet() or TCPGetArray() to read that data.*/
    /* TODO: if you want to send data, first check to see
    how much TX buffer is available with
    TCPIsPutReady(g_MyTelnetSock) and then use TCPput() or
    TCPputArray() to send data being careful to not send
    more data than TCPIsPutReady() told us was available.*/
    adcVal = read_adc();
    sprintf(potVal, "VDC = %f\r\n", adcVal*conVal);
    output_low(led0);
    bufSpace = TCPIsPutReady(g_MyTelnetSock);
    if(bufSpace >= sizeof(potVal))
        TCPputArray(g_MyTelnetSock, potVal, sizeof(potVal));
    delay_ms(2000);
}
```



```
VT 192.168.1.93 - Tera Term VT
File Edit Setup Control Window Help
VDC = 0.99
VDC = 0.99
VDC = 2.00
VDC = 2.00
VDC = 2.00
VDC = 2.00
VDC = 2.10
VDC = 2.20
VDC = 2.20
VDC = 2.20
VDC = 2.21
VDC = 2.21
VDC = 2.22
VDC = 2.23
VDC = 2.23
VDC = 2.23
VDC = 2.23
VDC = 2.23
VDC = 2.23
VDC = 2.29
VDC = 2.32
VDC = 2.32
VDC = 2.32
VDC = 2.32
```

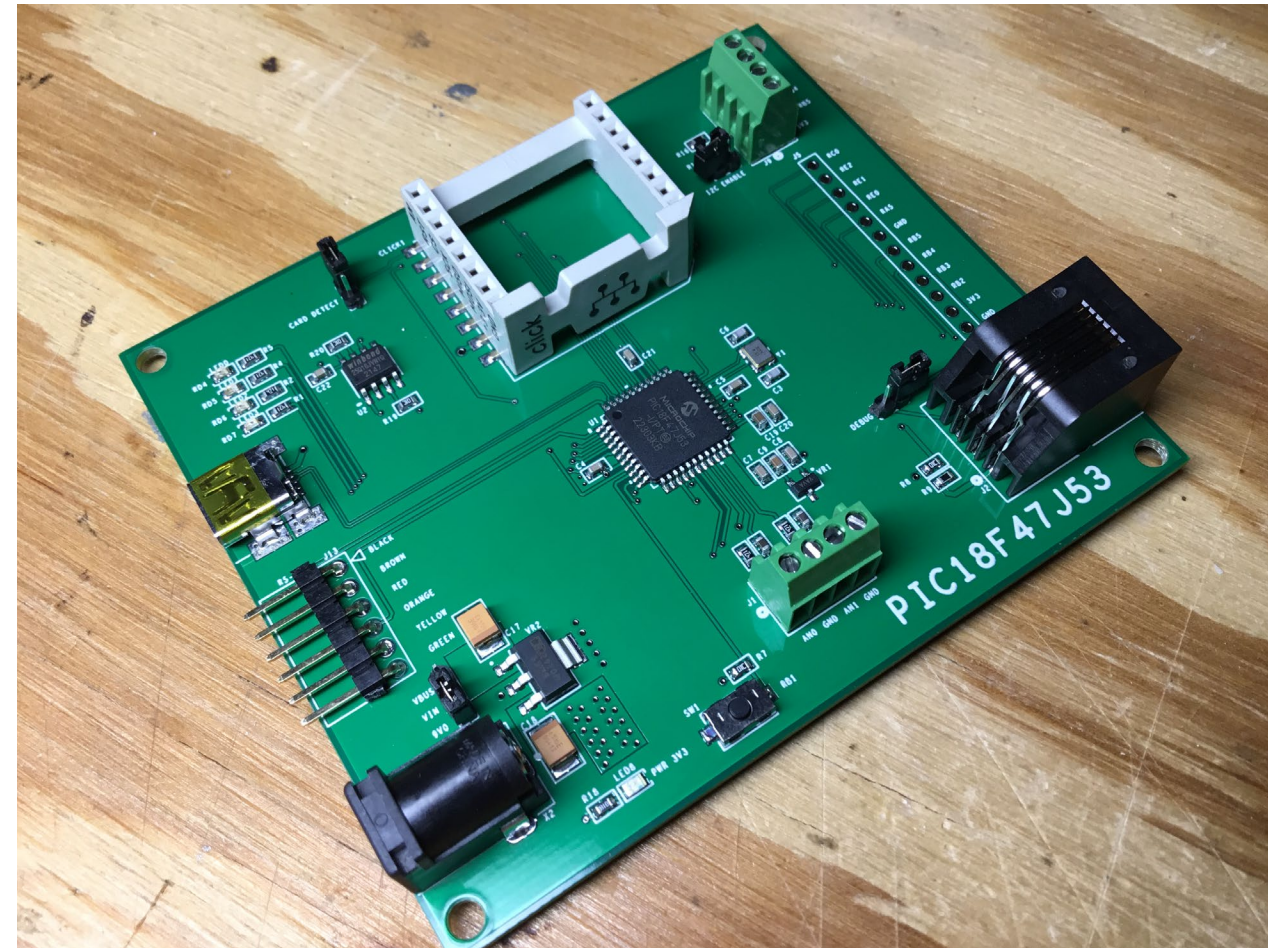


# Thank you for attending!!!

Please consider the resources below:

- [ccsinfo.com](http://ccsinfo.com)
- **CCS C Compiler Manual**
- **Master and Command C for PIC MCU (PDF)**

**MORE TO COME..**





**DesignNews**

Thank You

Sponsored by

