



**DesignNews**

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

**Day 1:**

**CCS C Project Wizardry 101**

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.

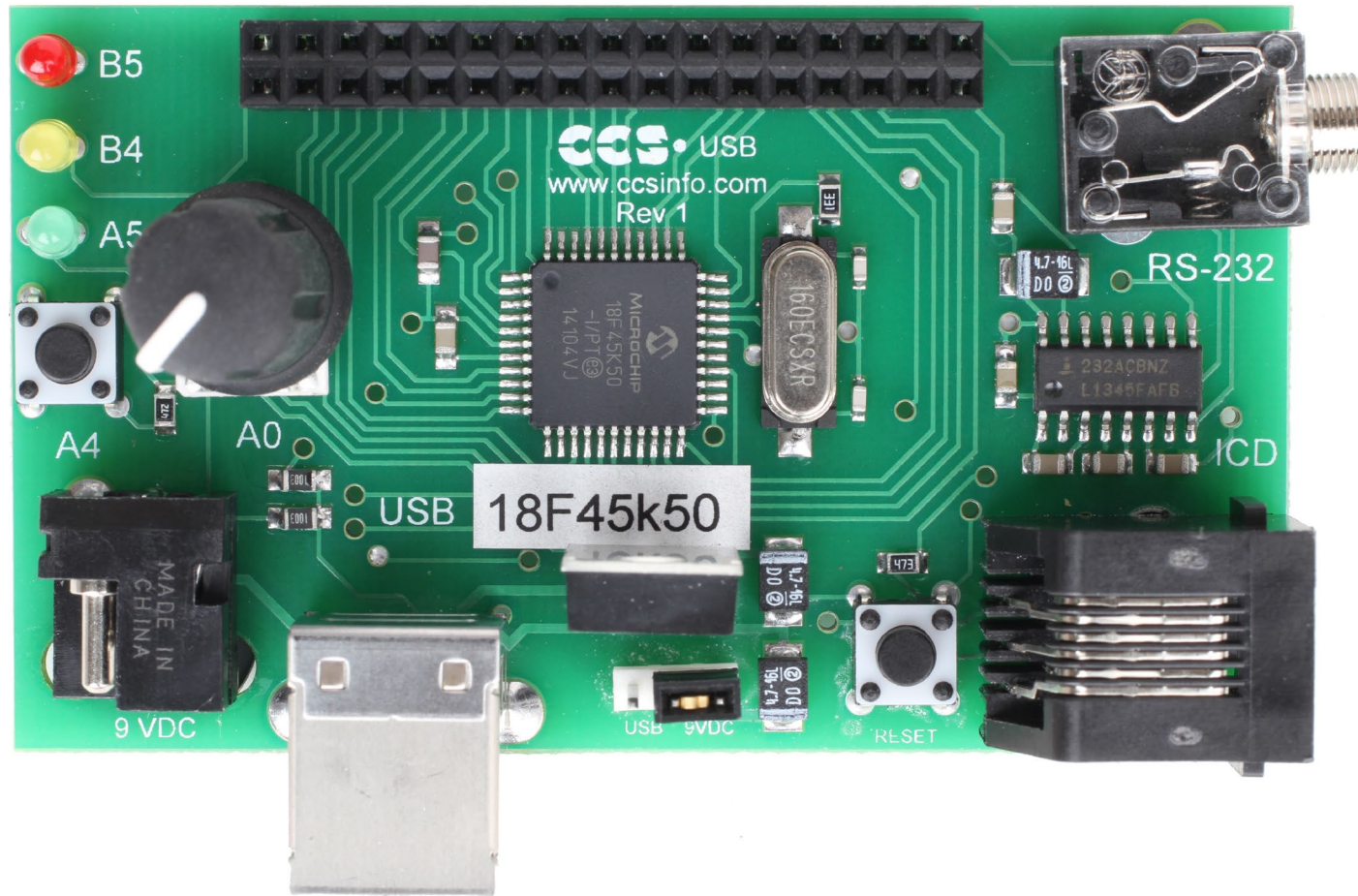


## Fred Eady

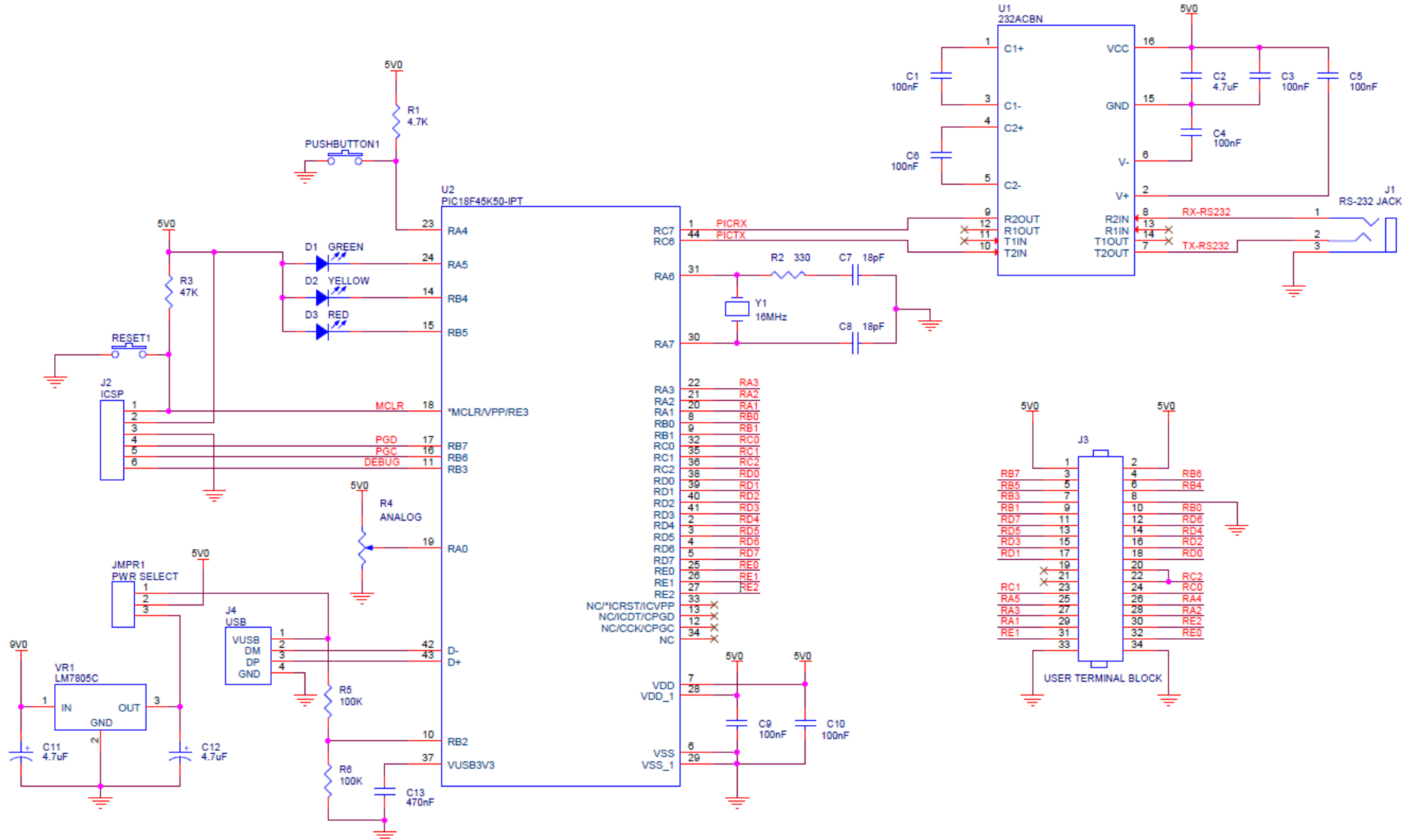
Visit 'Lecturer Profile' in your console for more details.

# AGENDA

- **Create a PIC18F45K50 Project**
- **Write the Application Code**
- **Compile and Run the Application Code**



# PIC18F45K50 Hardware



## Choose the PIC and Specify the CPU Clock Speed

Project Wizard - C:\Users\Public\cecCCS\day1\_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

General

Device

Family: PIC18  Debug Code

Device: PIC18F45K50 Fixed Compiler Version: None

Clock

Oscillator Type: Crystal  Use USB Low Speed

Crystal Clock Speed: 16 MHz  Use USB Full Speed

CPU Clock Speed: 16 MHz 4 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: PIC18F45K50 Frequency: 16,000,000

Create Project Cancel

## Define the Analog Pins

Project Wizard - C:\Users\Public\cecCCS\day1\_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

<input checked="" type="checkbox"/> A0	<input type="checkbox"/> C4
<input type="checkbox"/> A1	<input type="checkbox"/> C5
<input type="checkbox"/> A2	<input type="checkbox"/> C6
<input type="checkbox"/> A3	<input type="checkbox"/> C7
<input type="checkbox"/> A5	<input type="checkbox"/> D0
<input type="checkbox"/> E0	<input type="checkbox"/> D1
<input type="checkbox"/> E1	<input type="checkbox"/> D2
<input type="checkbox"/> E2	<input type="checkbox"/> D3
<input type="checkbox"/> B2	<input type="checkbox"/> D4
<input type="checkbox"/> B3	<input type="checkbox"/> D5
<input type="checkbox"/> B1	<input type="checkbox"/> D6
<input type="checkbox"/> B4	<input type="checkbox"/> D7
<input type="checkbox"/> B0	<input type="checkbox"/> All
<input type="checkbox"/> B5	
<input type="checkbox"/> C2	
<input type="checkbox"/> C3	

Range 0-Vdd

Units: 0-1023

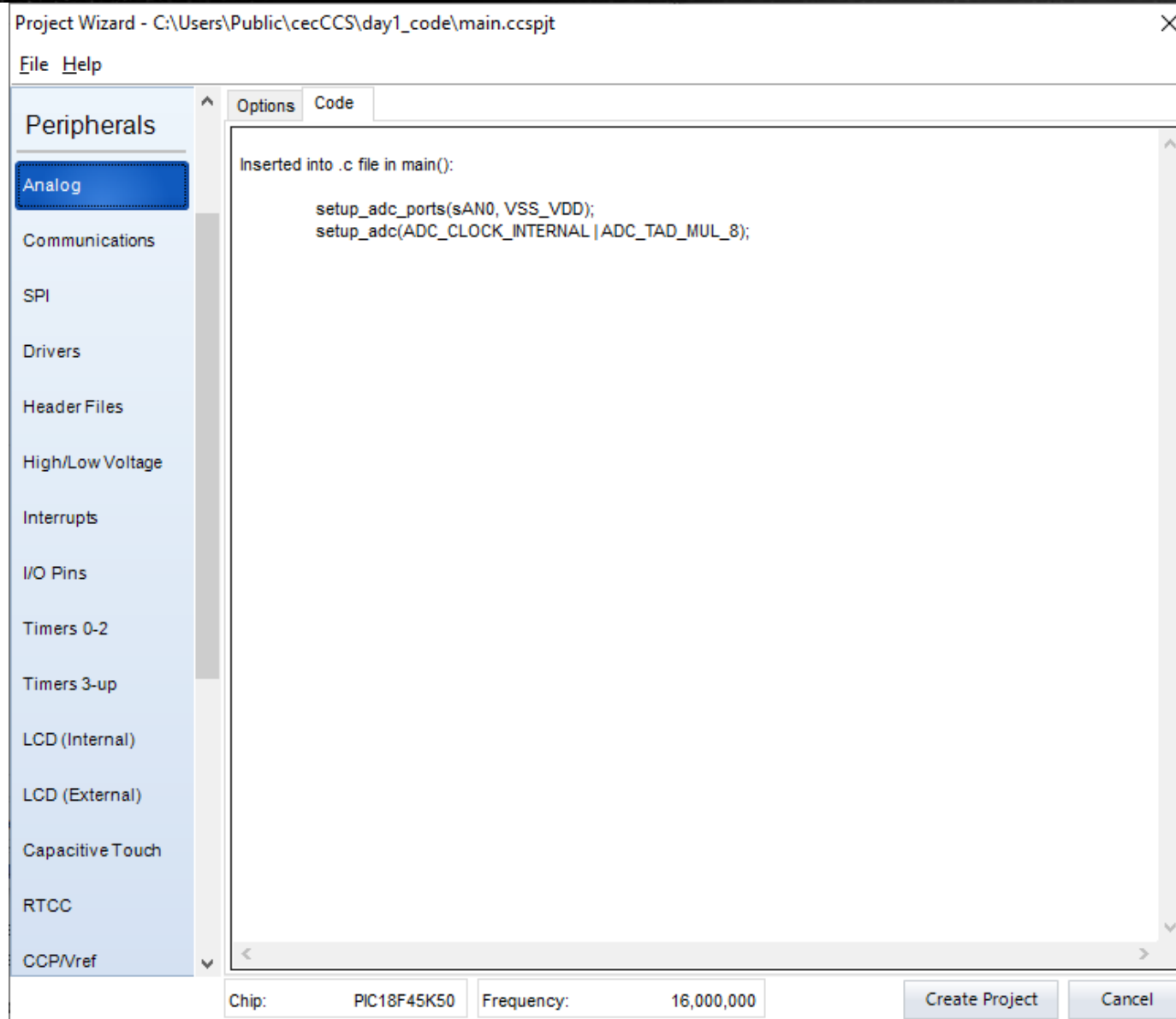
Internal RC Clock

Acquisition time: 1.6 us

Chip: PIC18F45K50 Frequency: 16,000,000

Create Project Cancel

## Peek at the ADC Setup Code



Project Wizard - C:\Users\Public\cecCCS\day1\_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/Vref

Options Code

Inserted into .c file in main():

```
setup_adc_ports(sAN0, VSS_VDD);  
setup_adc(ADC_CLOCK_INTERNAL | ADC_TAD_MUL_8);
```

Chip: PIC18F45K50 Frequency: 16,000,000 Create Project Cancel



## Configure the UART

Project Wizard - C:\Users\Public\cecCCS\day1\_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: 9600

Parity: None

Transmit Pin: C6

Receive Pin: C7

Invert

Float\_high

Errors

External interrupt

Enable Pin: None

Receive Enable Pin: None

Bits: 8

Stream: PORT1

Buffer Size: 0

ID: 10

I2C

Use I2C

SDA: None

SCL: None

Master  Slave

Fast  Slow

Restart WDT on I2C

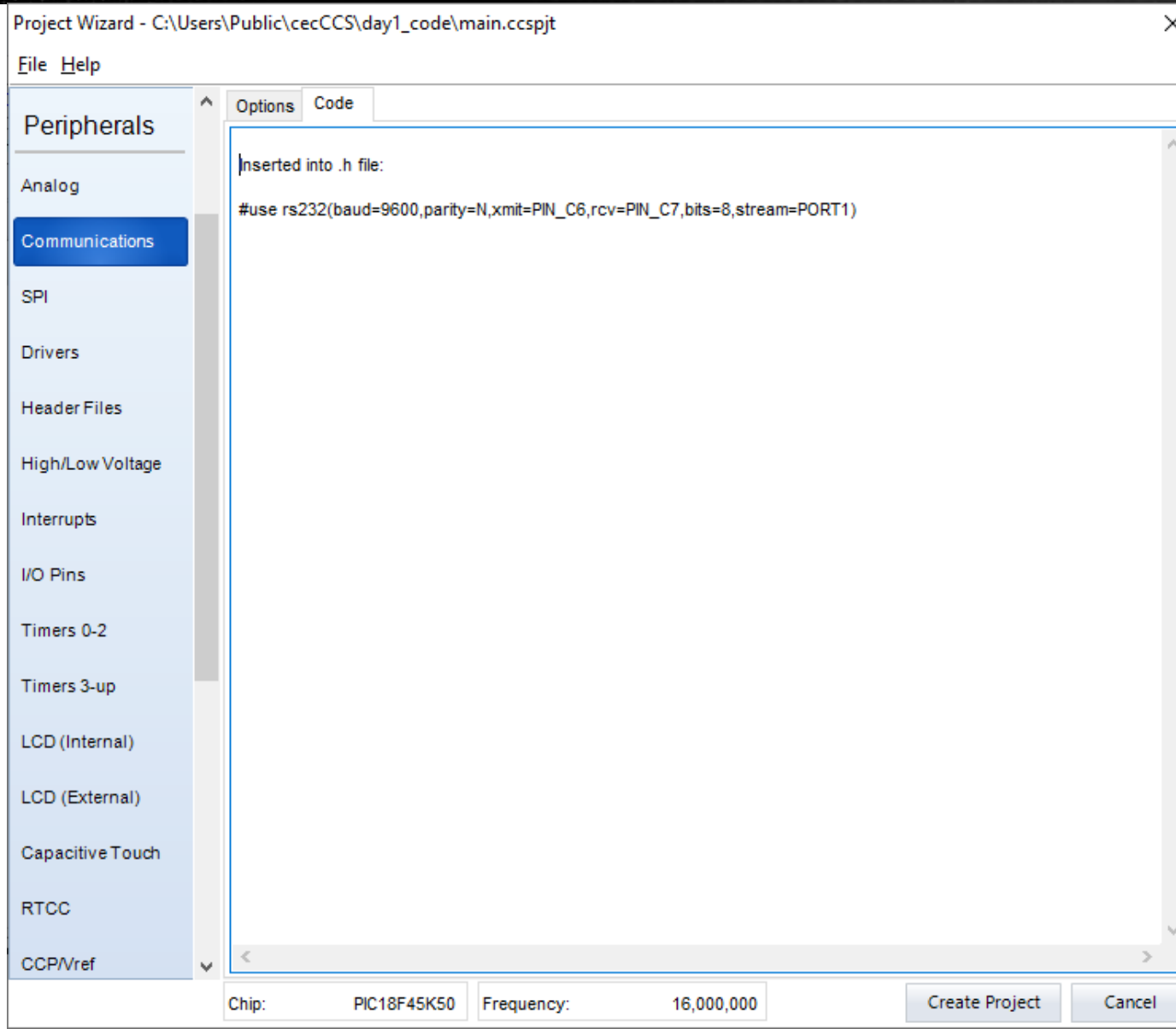
Force Hardware

Slave Address:

Chip: PIC18F45K50 Frequency: 16,000,000

Create Project Cancel

## Peek at the UART Initialization Code



Project Wizard - C:\Users\Public\cecCCS\day1\_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

```
Inserted into .h file:  
  
#use rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8,stream=PORT1)
```

Chip: PIC18F45K50 Frequency: 16,000,000 Create Project Cancel

## Define the GPIO Pins

Project Wizard - C:\Users\Public\cecCCS\day1\_code\main.ccsjpt

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

I/O Pins

Port C

PIN_C0	None	Name:	
PIN_C1	None	Name:	
PIN_C2	None	Name:	
PIN_C6	(RS232#1 XMIT)	Name:	
PIN_C7	(RS232#1 RCV)	Name:	

Port D

PIN_D0	None	Name:	
PIN_D1	None	Name:	
PIN_D2	None	Name:	
PIN_D3	None	Name:	
PIN_D4	None	Name:	
PIN_D5	None	Name:	
PIN_D6	None	Name:	
PIN_D7	None	Name:	

Port E

PIN_E0	None	Name:	
PIN_E1	None	Name:	
PIN_E2	None	Name:	
PIN_E3	None	Name:	

Pull-up Resistors

Chip: PIC18F45K50 Frequency: 16,000,000

Create Project Cancel

## Define the GPIO Pins

Project Wizard - C:\Users\Public\cecCCS\day1\_code\main.ccsjpt

File Help

Options Code

I/O Pins

High/Low Voltage

Interrupts

Timers 0-2

Timers 3-up

LCD (Internal)

LCD (External)

Capacitive Touch

RTCC

CCP/Vref

Comparator

USB

TCP/IP

MODBus

CANBus

Bootloader

**I/O Pins**

Port A

PIN_A0	(Analog)	Name:	
PIN_A1	None	Name:	
PIN_A2	None	Name:	
PIN_A3	None	Name:	
PIN_A4	Input	Name:	btnA4
PIN_A5	Output	Name:	ledGrnA5
PIN_A6	None	Name:	
PIN_A7	None	Name:	

Port B

PIN_B0	None	Name:	
PIN_B1	None	Name:	
PIN_B2	None	Name:	
PIN_B3	None	Name:	
PIN_B4	Output	Name:	ledYelB4
PIN_B5	Output	Name:	ledRedB5
PIN_B6	None	Name:	
PIN_B7	None	Name:	

Pull-up Resistors

Port C

PIN_C0	None	Name:	
PIN_C1	None	Name:	

Chip: PIC18F45K50 Frequency: 16,000,000 Create Project Cancel

## Set and Expose the Fuses

Project Wizard - C:\Users\Public\cecCCS\day1\_code\main.ccsproj

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

Primary clock is system clock when scs=00

Fail-safe clock monitor enabled

Internal External Switch Over mode enabled

Power Up Timer

No brownout reset

Brownout reset at 1.9V

Low-Power Brownout reset is enabled

PORTB pins are configured as analog input channels on RESET

T3 Clock In is on C0

SDO is on RB3

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

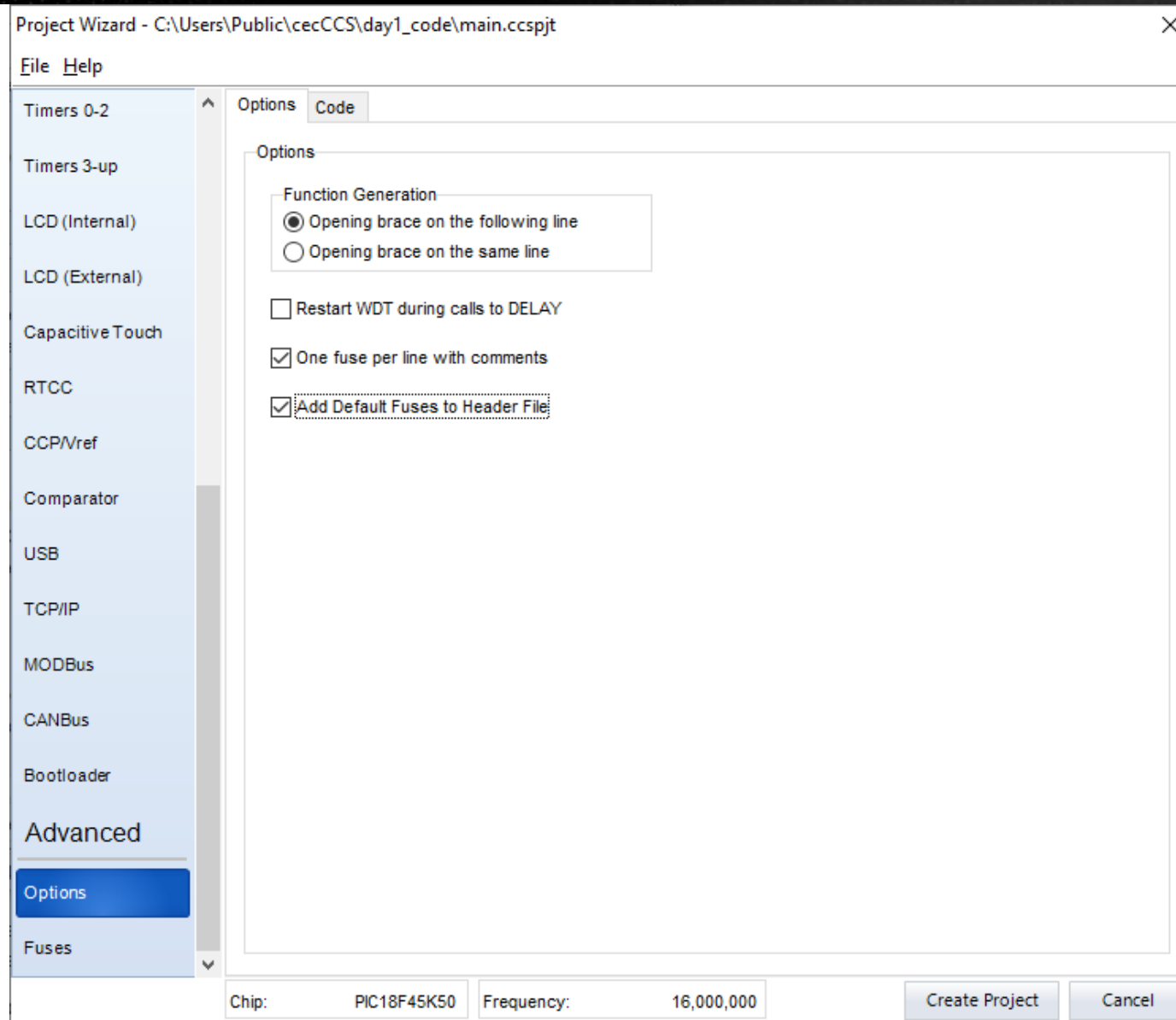
Configuration registers write protected

Boot block write protected

Chip: PIC18F45K50 Frequency: 16,000,000

Create Project Cancel

## Set and Expose the Fuses and Specify Function Brace Positions

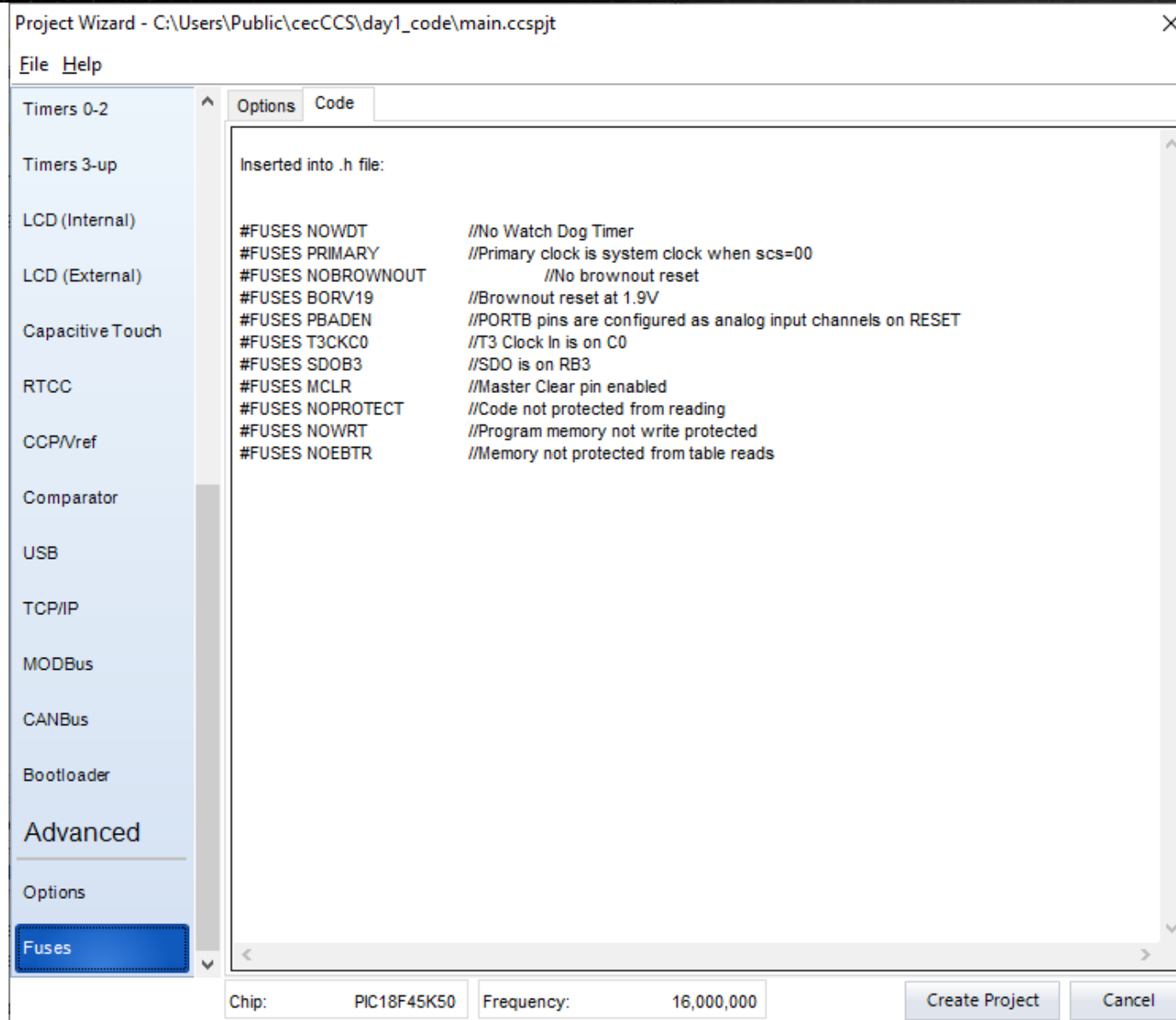


The screenshot shows the 'Options' tab of the CCS C Project Wizard. The left sidebar lists various project options, with 'Options' selected. The main area contains the following settings:

- Function Generation:**
  - Opening brace on the following line
  - Opening brace on the same line
- Restart WDT during calls to DELAY
- One fuse per line with comments
- Add Default Fuses to Header File

At the bottom of the dialog, the 'Chip' is set to 'PIC18F45K50' and the 'Frequency' is set to '16,000,000'. There are 'Create Project' and 'Cancel' buttons at the bottom right.

## Expose the Fuses and Create the Project



Project Wizard - C:\Users\Public\cecCCS\day1\_code\main.ccsjpt

File Help

Options Code

Inserted into .h file:

```
#FUSES NOWDT           //No Watch Dog Timer
#FUSES PRIMARY         //Primary clock is system clock when scs=00
#FUSES NOBROWNOUT     //No brownout reset
#FUSES BORV19         //Brownout reset at 1.9V
#FUSES PBAEN          //PORTB pins are configured as analog input channels on RESET
#FUSES T3CKC0         //T3 Clock In is on C0
#FUSES SDOB3          //SDO is on RB3
#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
```

Chip: PIC18F45K50 Frequency: 16,000,000 Create Project Cancel

## main.h

```
#include <18F45K50.h>
#device ADC=10

#FUSES NOWDT           //No Watch Dog Timer
#FUSES PRIMARY        //Primary clock is system clock when scs=00
#FUSES NOBROWNOUT     //No brownout reset
#FUSES BORV19         //Brownout reset at 1.9V
#FUSES PBADEN         //PORTB pins are configured as analog input channels on RESET
#FUSES T3CKC0        //T3 Clock In is on C0
#FUSES SDOB3         //SD0 is on RB3
#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=16MHz)
#use FIXED_IO( A_outputs=PIN_A5 )
#use FIXED_IO( B_outputs=PIN_B5,PIN_B4 )
#define btnA4      PIN_A4
#define ledGrnA5   PIN_A5
#define ledYelB4   PIN_B4
#define ledRedB5   PIN_B5

#use rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8,stream=PORT1)
```



## Initialization Code

```
#include <main.h>

unsigned int8 biteIn;
unsigned int1 btnPressed;
long adcVal;
float conVal = 0.004887586;

void main()
{
    setup_adc_ports(sAN0, VSS_VDD);
    setup_adc(ADC_CLOCK_INTERNAL | ADC_TAD_MUL_8);
    set_adc_channel(0);
    btnPressed = 0;
    output_high(ledRedB5);
    output_high(ledYelB4);
    output_high(ledGrnA5);
    printf("ALL LEDS = OFF\r\n");
}
```



## Button Sense Code

```
while(TRUE)
{
    biteIn = input(btnA4);
    if(biteIn == 0)
    {
        delay_ms(10);
        biteIn = input(btnA4);
        if(biteIn == 0)
        {
            printf("Button is depressed..\r\n");
            btnPressed = 1;
        }
    }
    do
    {
        biteIn = input(btnA4);
        if(biteIn == 1)
        {
            delay_ms(10);
            biteIn = input(btnA4);
            if(biteIn == 1)
            {
                btnPressed = 0;
                printf("Button is released..\r\n");
            }
        }
    }
}while(btnPressed == 1);
```



## Analog-to-Digital Code

```
long adcVal;  
float conVal = 0.004887586;  
  
adcVal = read_adc();  
printf("adcVal = 0x%LX -> %lud\r\n", adcVal, adcVal);  
printf("adc voltage = %f\r\n", adcVal*conVal);
```

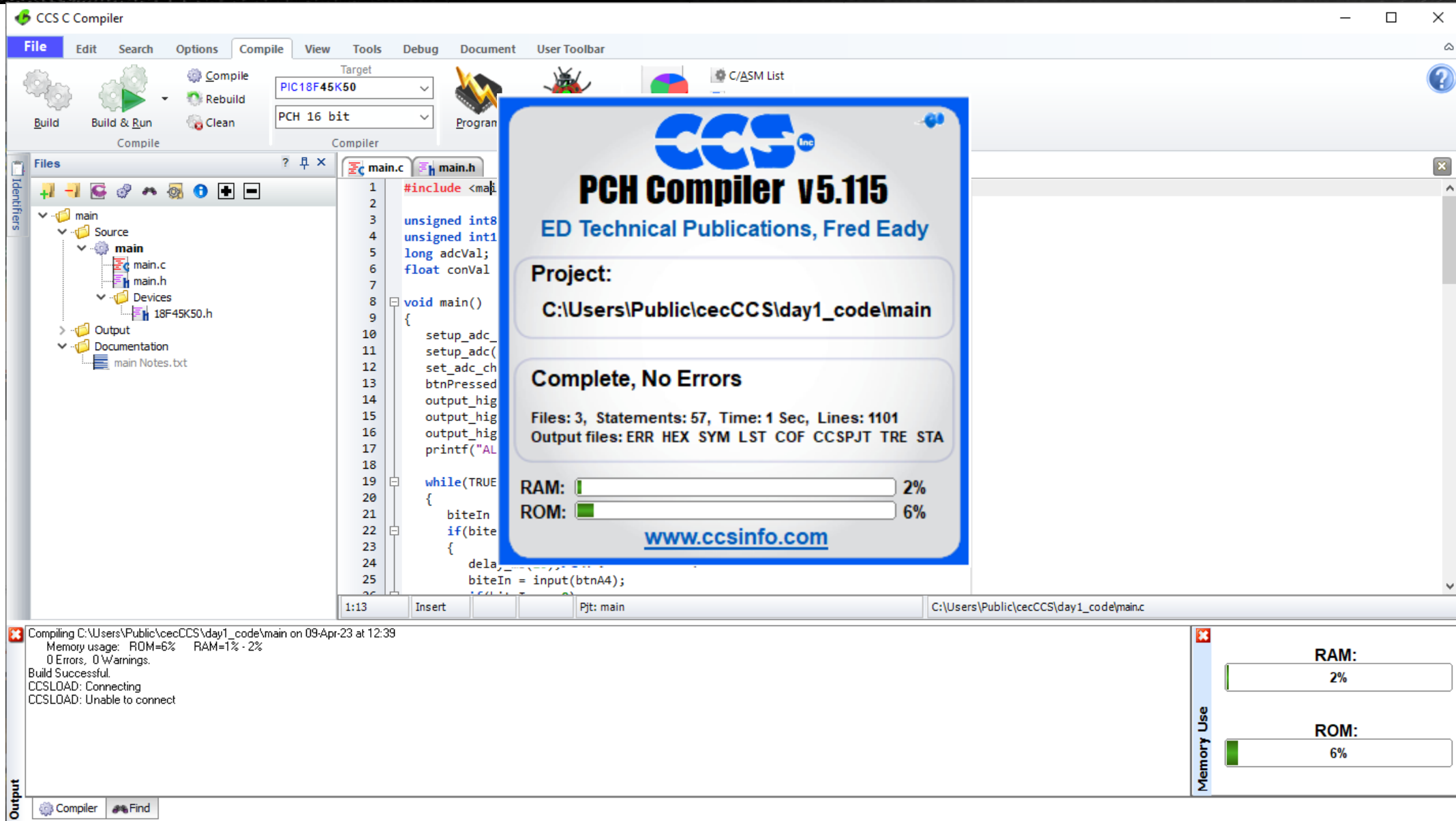


## LED Control Code

```
if(kbhit())
{
    biteIn = getc();
    switch(biteIn)
    {
        case 'R':
            output_low(ledRedB5);
            output_high(ledYelB4);
            output_high(ledGrnA5);
            printf("RED LED = ON\r\n");
            break;
        case 'Y':
            output_high(ledRedB5);
            output_low(ledYelB4);
            output_high(ledGrnA5);
            printf("YELLOW LED = ON\r\n");
            break;
        case 'G':
            output_high(ledRedB5);
            output_high(ledYelB4);
            output_low(ledGrnA5);
            printf("GREEN LED = ON\r\n");
            break;
        default:
            output_high(ledRedB5);
            output_high(ledYelB4);
            output_high(ledGrnA5);
            printf("Invalid Character Received -> All LEDs = OFF\r\n");
            break;
    }
}
```



# Compile the Application Code



CCS C Compiler

File Edit Search Options Compile View Tools Debug Document User Toolbar

Build Build & Run Clean

Target: PIC18F45K50  
Compiler: PCH 16 bit

Files

- main
  - Source
    - main
      - main.c
      - main.h
    - Devices
      - 18F45K50.h
    - Output
    - Documentation
      - main Notes.txt

```
1 #include <main.h>
2
3 unsigned int8
4 unsigned int1
5 long adcVal;
6 float conVal;
7
8 void main()
9 {
10     setup_adc_
11     setup_adc(
12     set_adc_ch
13     btnPressed
14     output_hig
15     output_hig
16     output_hig
17     printf("AL
18
19     while(TRUE
20     {
21         biteIn
22         if(bite
23         {
24             dela
25             biteIn = input(btnA4);
```

1:13 Insert Pjt: main C:\Users\Public\cecCCS\day1\_code\main.c

RAM: 2%  
ROM: 6%

www.ccsinfo.com

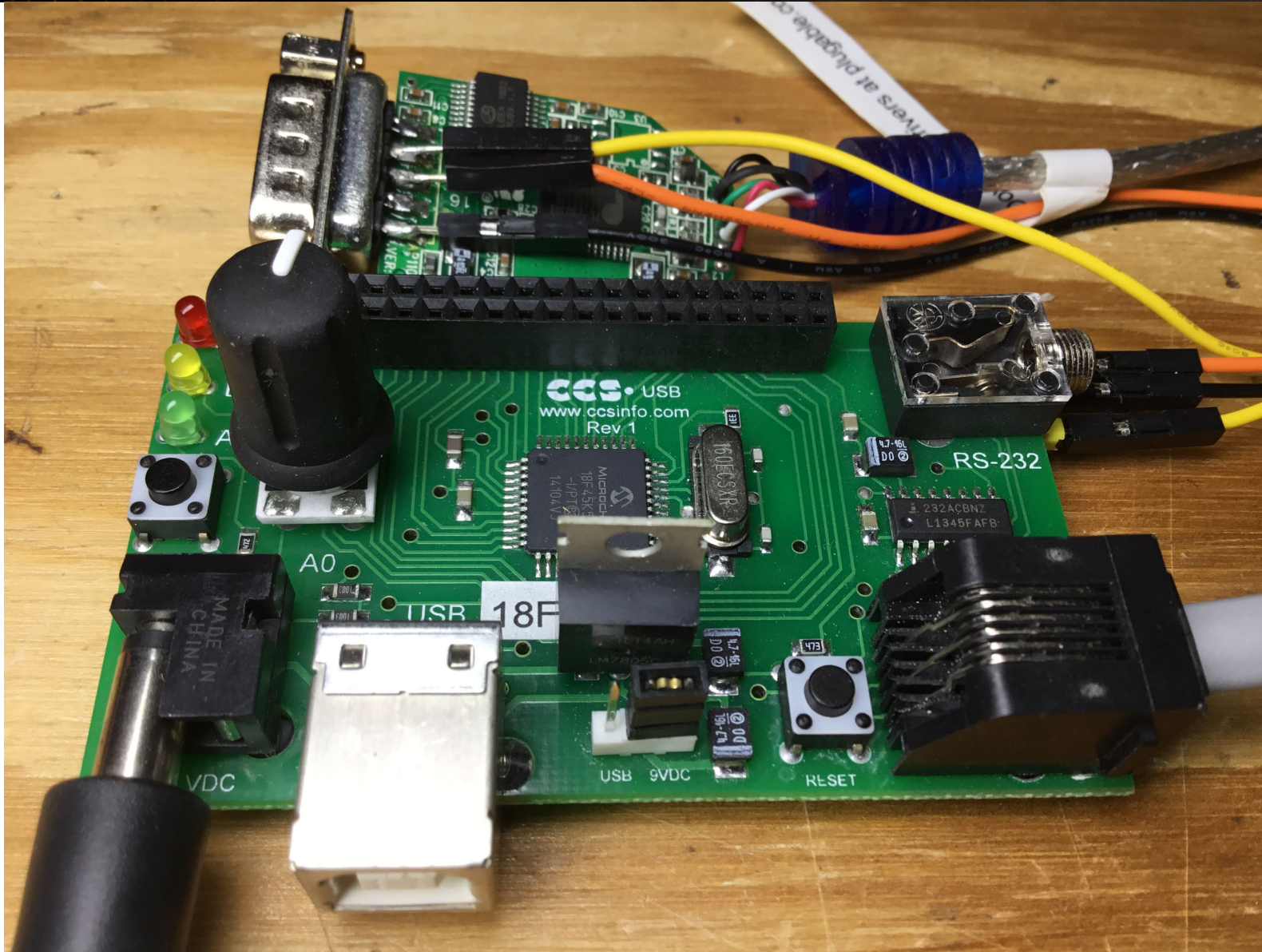
Output

Compiling C:\Users\Public\cecCCS\day1\_code\main on 09-Apr-23 at 12:39  
Memory usage: ROM=6% RAM=1% - 2%  
0 Errors, 0 Warnings.  
Build Successful.  
CCSLOAD: Connecting  
CCSLOAD: Unable to connect

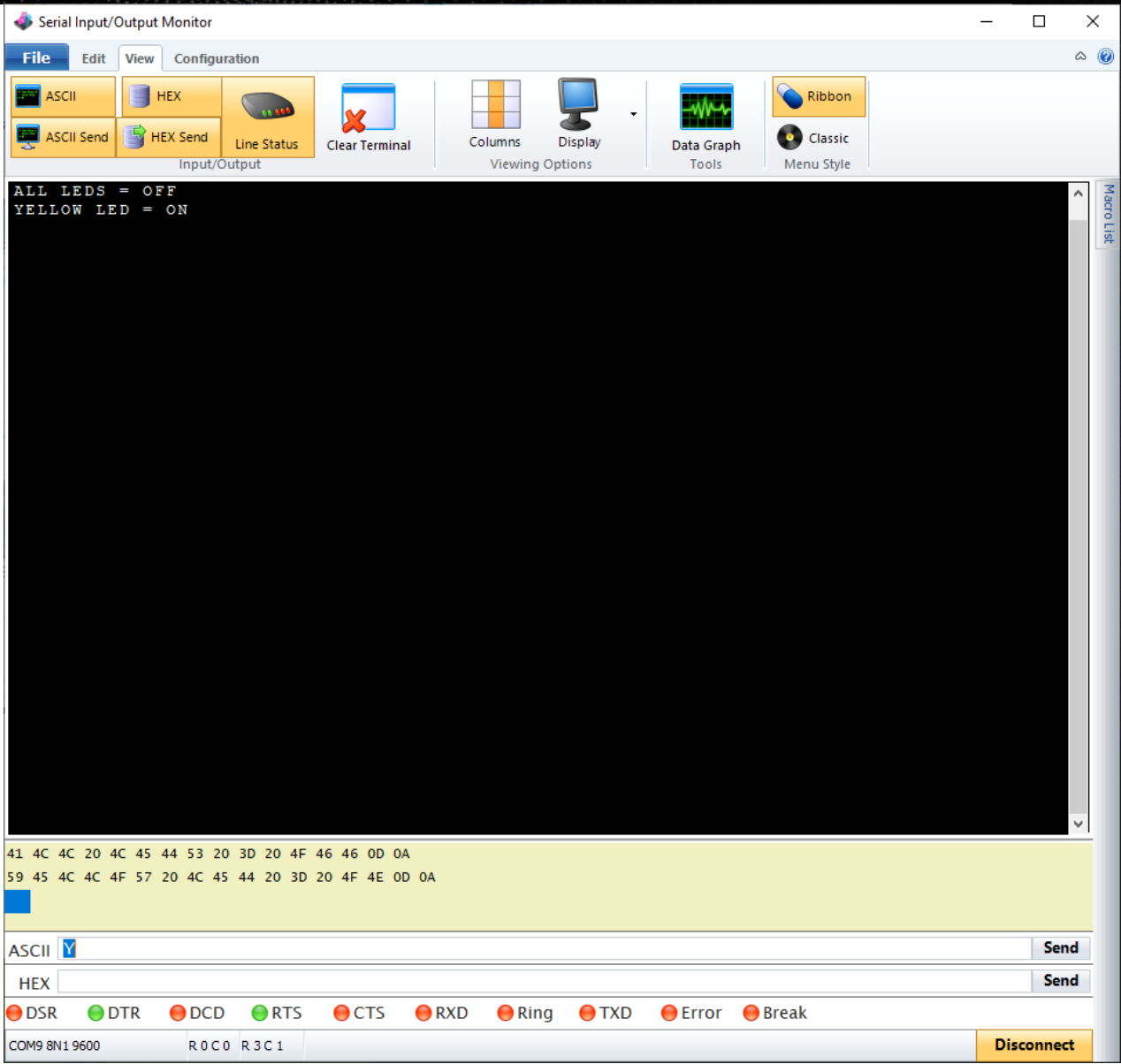
Memory Use

RAM: 2%  
ROM: 6%

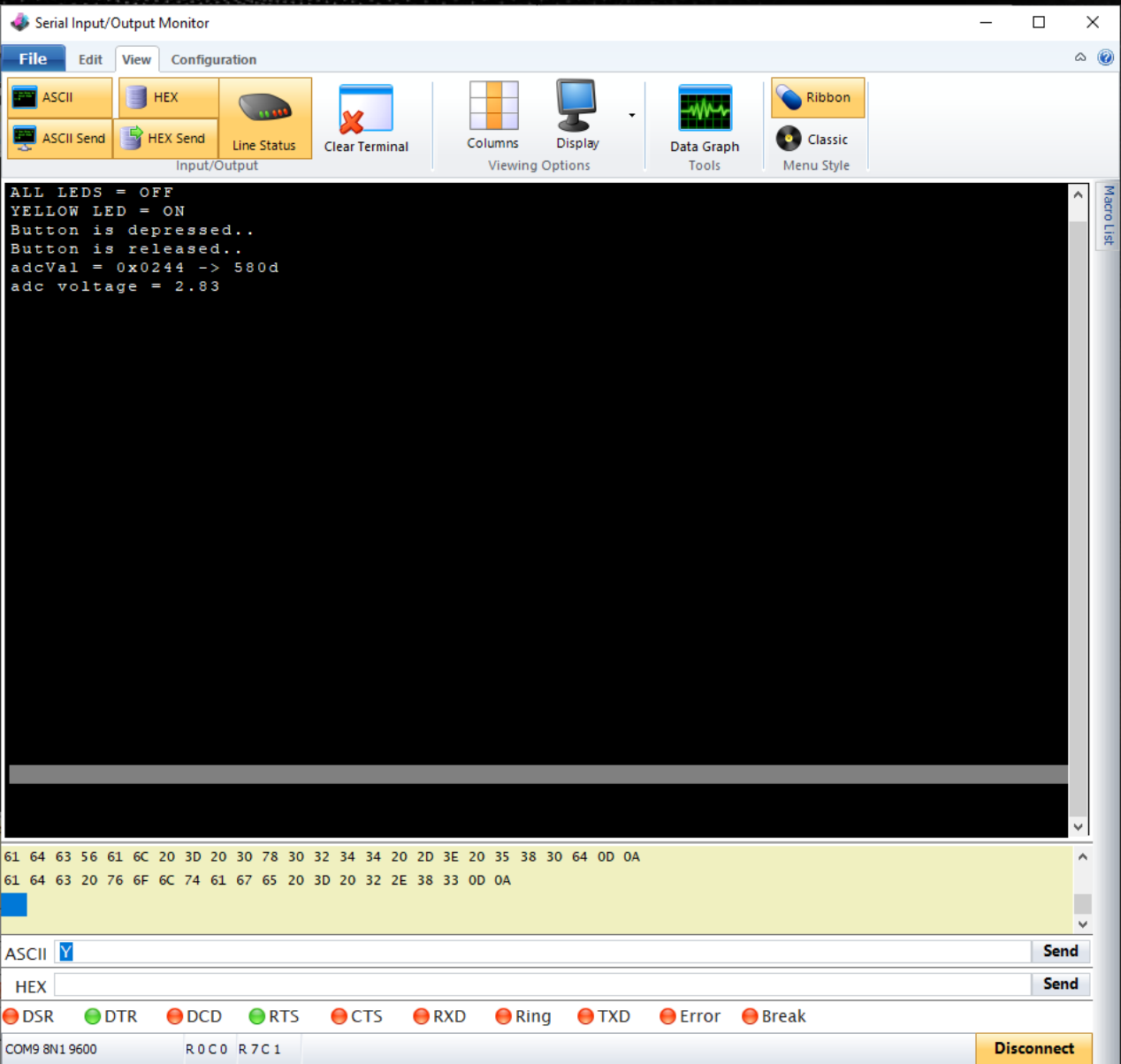
## Run the Application Code



# Run the Application Code



# Run the Application Code



The screenshot shows the Serial Input/Output Monitor application window. The main terminal area displays the following text:

```
ALL LEDES = OFF  
YELLOW LED = ON  
Button is depressed..  
Button is released..  
adcVal = 0x0244 -> 580d  
adc voltage = 2.83
```

Below the terminal, there are two lines of hexadecimal data:

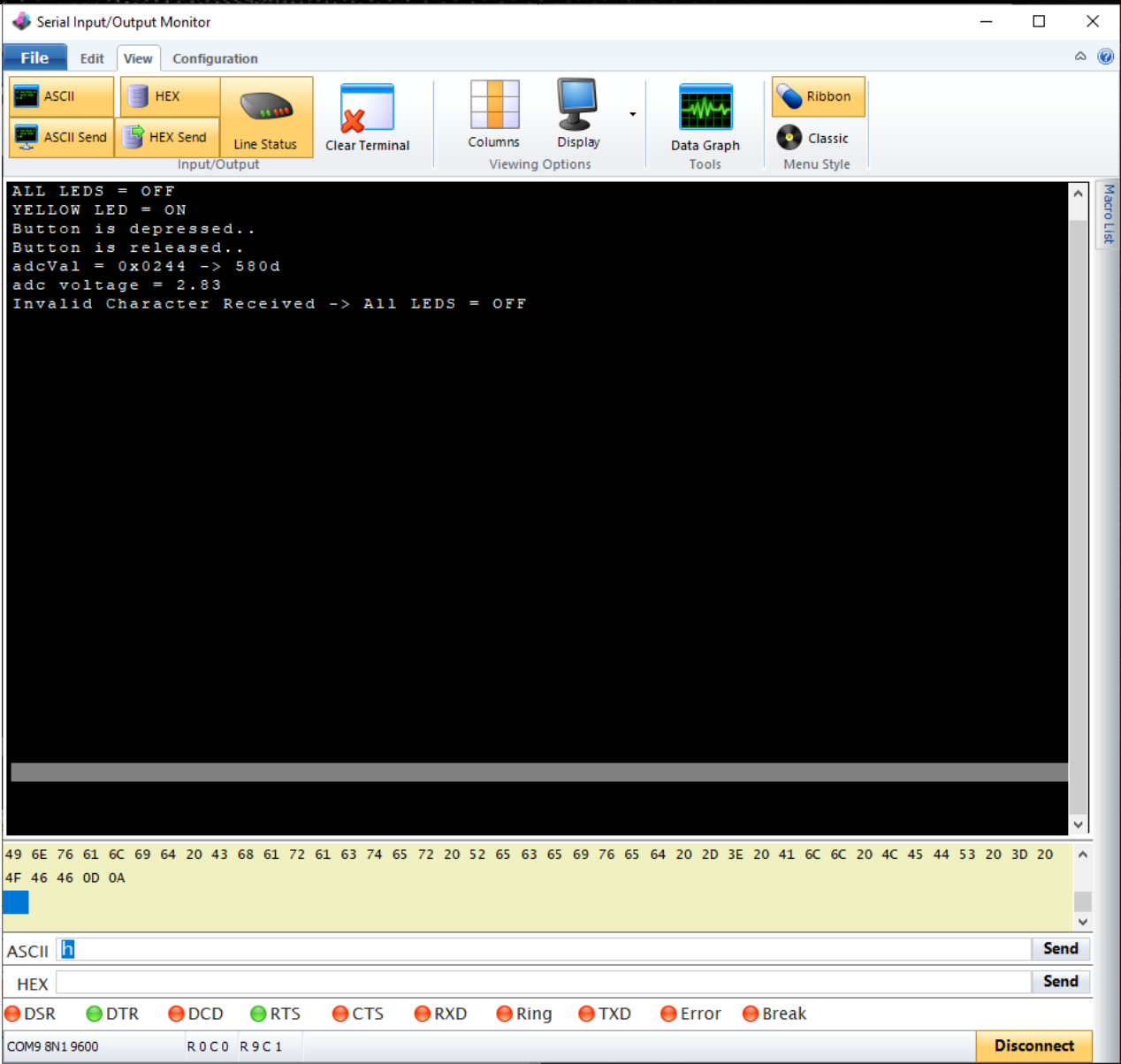
```
61 64 63 56 61 6C 20 3D 20 30 78 30 32 34 34 20 2D 3E 20 35 38 30 64 0D 0A  
61 64 63 20 76 6F 6C 74 61 67 65 20 3D 20 32 2E 38 33 0D 0A
```

The interface includes a menu bar (File, Edit, View, Configuration), a ribbon with various tools like ASCII, HEX, Line Status, Clear Terminal, Columns, Display, Data Graph, and Menu Style, and a status bar at the bottom showing COM9 8N1 9600 and a Disconnect button.





# Run the Application Code



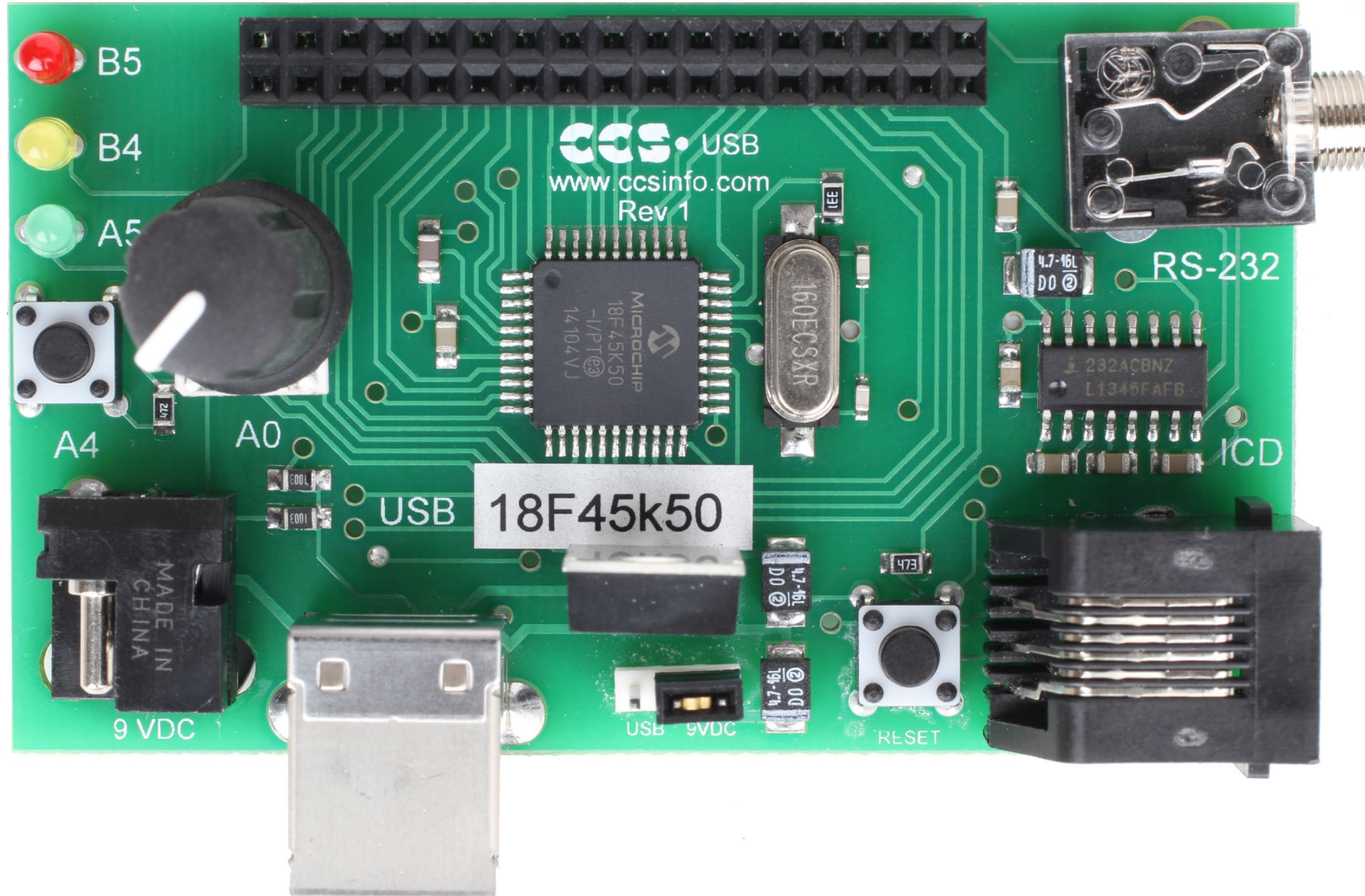
The screenshot shows the Serial Input/Output Monitor application window. The terminal displays the following output:

```
ALL LEDS = OFF  
YELLOW LED = ON  
Button is depressed..  
Button is released..  
adcVal = 0x0244 -> 580d  
adc voltage = 2.83  
Invalid Character Received -> All LEDS = OFF
```

The interface includes a menu bar (File, Edit, View, Configuration), a ribbon with various tools like ASCII, HEX, Line Status, Clear Terminal, Columns, Display, Data Graph, and Menu Style, and a status bar at the bottom with control buttons like DSR, DTR, DCD, RTS, CTS, RXD, Ring, TXD, Error, Break, and a Disconnect button.



# Bonus USB Version

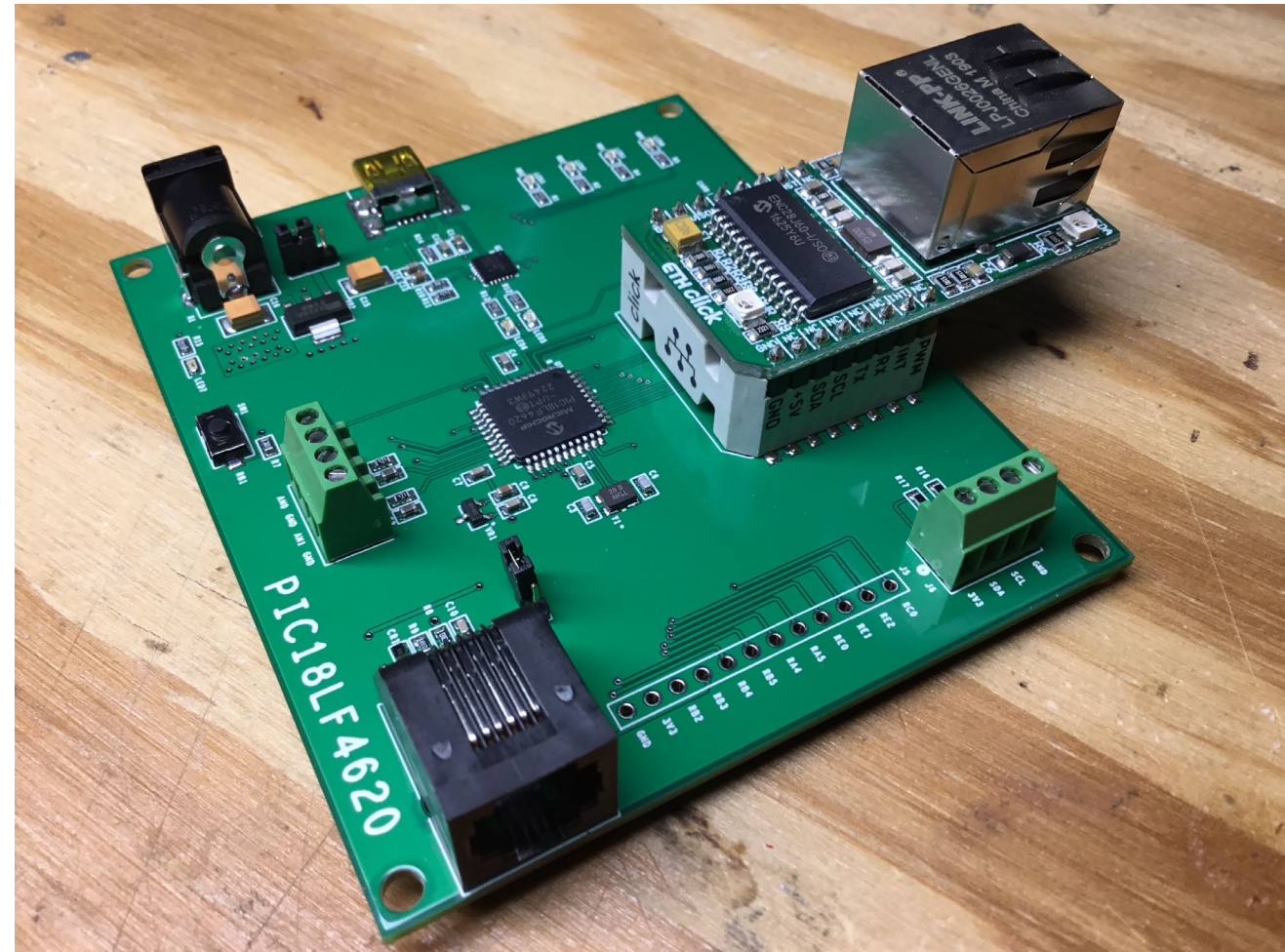


# Thank you for attending!!!

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

