



DesignNews

PIC Microcontroller Embedded Development Using the CCS PIC MCU C Compiler

Day 3:

USB 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.

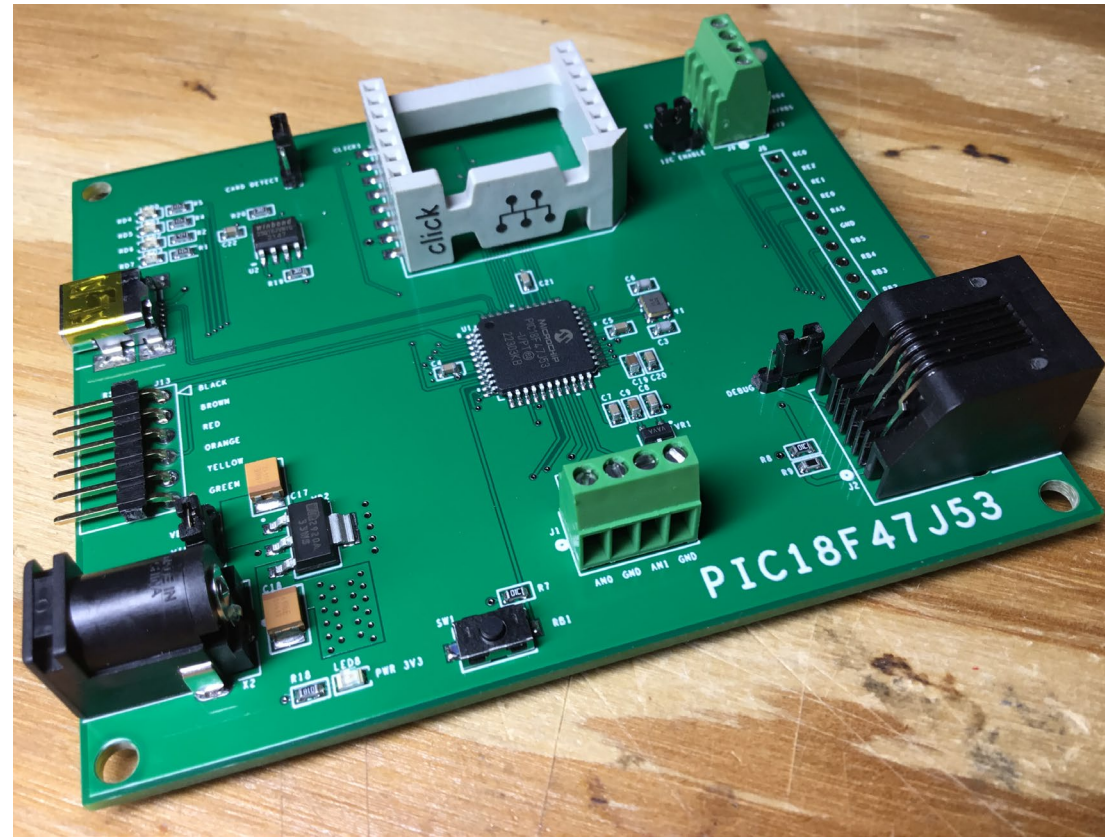


Fred Eady

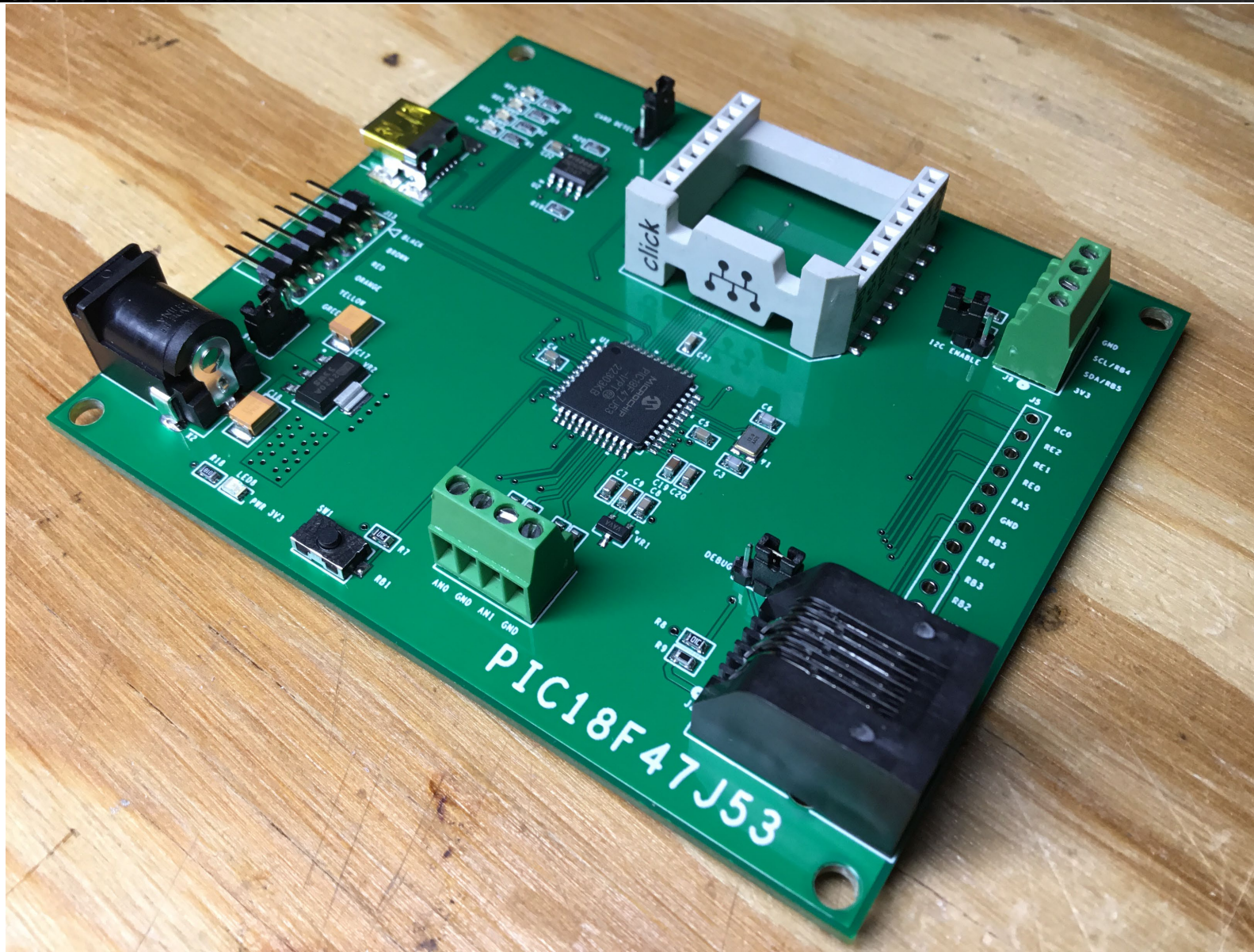
Visit 'Lecturer Profile' in your console for more details.

AGENDA

- **Create a PIC18F47J53 USB Project**
- **Write the Application Code**
- **Code the USB Interface Application**
- **Run the USB Control Application**



PIC18F47J53 Hardware



Choose the PIC and Specify the CPU Clock Speed

Project Wizard - C:\Users\Public\cecCCS\day3_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

General

Device

Family: PIC18 Debug Code

Device: PIC18F47J53 Fixed Compiler Version: None

Clock

Oscillator Type: Crystal Use USB Low Speed

Crystal Clock Speed: 12 MHz Use USB Full Speed

CPU Clock Speed: 48 MHz 12 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: PIC18F47J53 Frequency: 48,000,000

Create Project Cancel

Define the Analog Pins

Project Wizard - C:\Users\Public\cecCCS\day3_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

- B0
- C2
- B1
- B3
- B2
- E2
- E1
- E0
- A5
- A3
- A2
- A1
- A0
- All

Range Vref-Vref

Units: 0-4095

Internal RC Clock

Acquisition time: 1.6 us

Chip: PIC18F47J53 Frequency: 48,000,000

Create Project Cancel

Configure the UART

Project Wizard - C:\Users\Public\cecCCS\day3_code\main.ccspt

File Help

Device Selection

Device

Example 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)

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: B5

SCL: B4

Master Slave

Fast Slow

Restart WDT on I2C

Force Hardware

Slave Address:

Chip: PIC18F47J53 Frequency: 48,000,000

Create Project Cancel

Configure the USB Engine

Project Wizard - C:\Users\Public\cecCCS\day3_code\main.ccspt

File Help

High/Low Voltage
Interrupts
I/O Pins
Timers 0-2
Timers 3-up
LCD (Internal)
LCD (External)
Capacitive Touch
RTCC
CCP/Ref
Comparator
USB
TCP/IP
MODBus
CANBus
Bootloader

Options Code

USB

Enable USB

Software Algorithm: Interrupts

Connection Sense Pin: None

Vendor ID (VID): 0x

Product ID (PID): 0x

Manufacturer String:

Product String:

Maximum Bus Power (mA): 500

Device Class:

- Communication Device Class (CDC, Virtual COM port)
- Human Interface Device Class (HID)
- HID and CDC
- Bulk USB

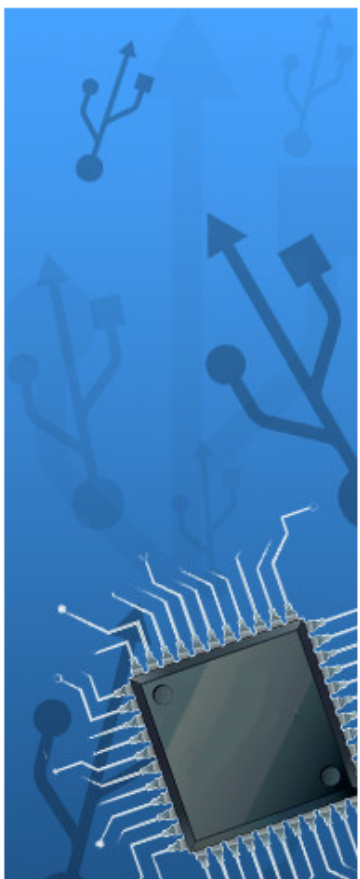
PIC Transmit Size (bytes): 0

PIC Receive Size (bytes): 0

Enter Tx and Rx size fom 0 to 63

Chip: PIC18F47J53 Frequency: 48,000,000

Create Project Cancel



Set and Expose the Fuses and Create the Project

Project Wizard - C:\Users\Public\cecCCS\day3_code\main.ccsjpt

File Help

Timers 0-2
Timers 3-up
LCD (Internal)
LCD (External)
Capacitive Touch
RTCC
CCP/ref
Comparator
USB
TCP/IP
MODBus
CANBus
Bootloader
Advanced
Options
Fuses

Options Code

Fuses

- Include Fuses
- Stack full/underflow will cause reset
- Extended set extension and Indexed Addressing mode enabled
- Code protected from reads
- High-power SOSC circuit is selected
- Output clock on OSC2
- Fail-safe clock monitor enabled
- Internal External Switch Over mode enabled
- DSWDT uses INTRC as reference clock
- RTCC uses Secondary Oscillator as reference source
- BOR enabled in Deep Sleep
- Deep Sleep Watchdog Timer enabled
- DSWDT_*
- Allows only one reconfiguration of peripheral pins
- ADC is 12-bits
- MSSP uses 7 bit Masking mode
- Write/Erase Protect Page Start/End Location, set to last page or use WFPF=x to set page
- Configuration Words page is erase/write-protected
- All Flash memory may be erased or written
- Flash pages WFPF to Configuration Words page are write/erase protected

Chip: PIC18F47J53 Frequency: 48,000,000

Create Project Cancel



main.h

```
#include <18F47J53.h>
#device ADC=12

#FUSES NOWDT           //No Watch Dog Timer
#FUSES SOSC_HIGH       //High-power SOSC circuit is selected
#FUSES DSWDTOSC_INT    //DSWDT uses INTRC as reference clock
#FUSES RTCOSC_T1       //RTCC uses Secondary Oscillator as reference source
#FUSES IOL1WAY         //Allows only one reconfiguration of peripheral pins
#FUSES ADC12           //ADC is 12-bits
#FUSES MSSPMSK7        //MSSP uses 7 bit Masking mode
#FUSES WFPF            //Write/Erase Protect Page Start/End Location, set to last page or use WFPF=x to set page
#FUSES WPDIS           //All Flash memory may be erased or written
#FUSES WPEND           //Flash pages WFPF to Configuration Words page are write/erase protected

#use delay(clock=48MHz,crystal=12MHz,USB_FULL)
#use FIXED_IO( D_outputs=PIN_D7,PIN_D6,PIN_D5,PIN_D4 )
#define btnB1    PIN_B1
#define led0_ORG PIN_D4
#define led1_BLU PIN_D5
#define led2_RED PIN_D6
#define led3_GRN PIN_D7

#use rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8,stream=PORT1)
#use i2c(Master,Fast,sda=PIN_B5,scl=PIN_B4)

#define USB_CONFIG_BUS_POWER 500
#include <usb_cdc.h>
```

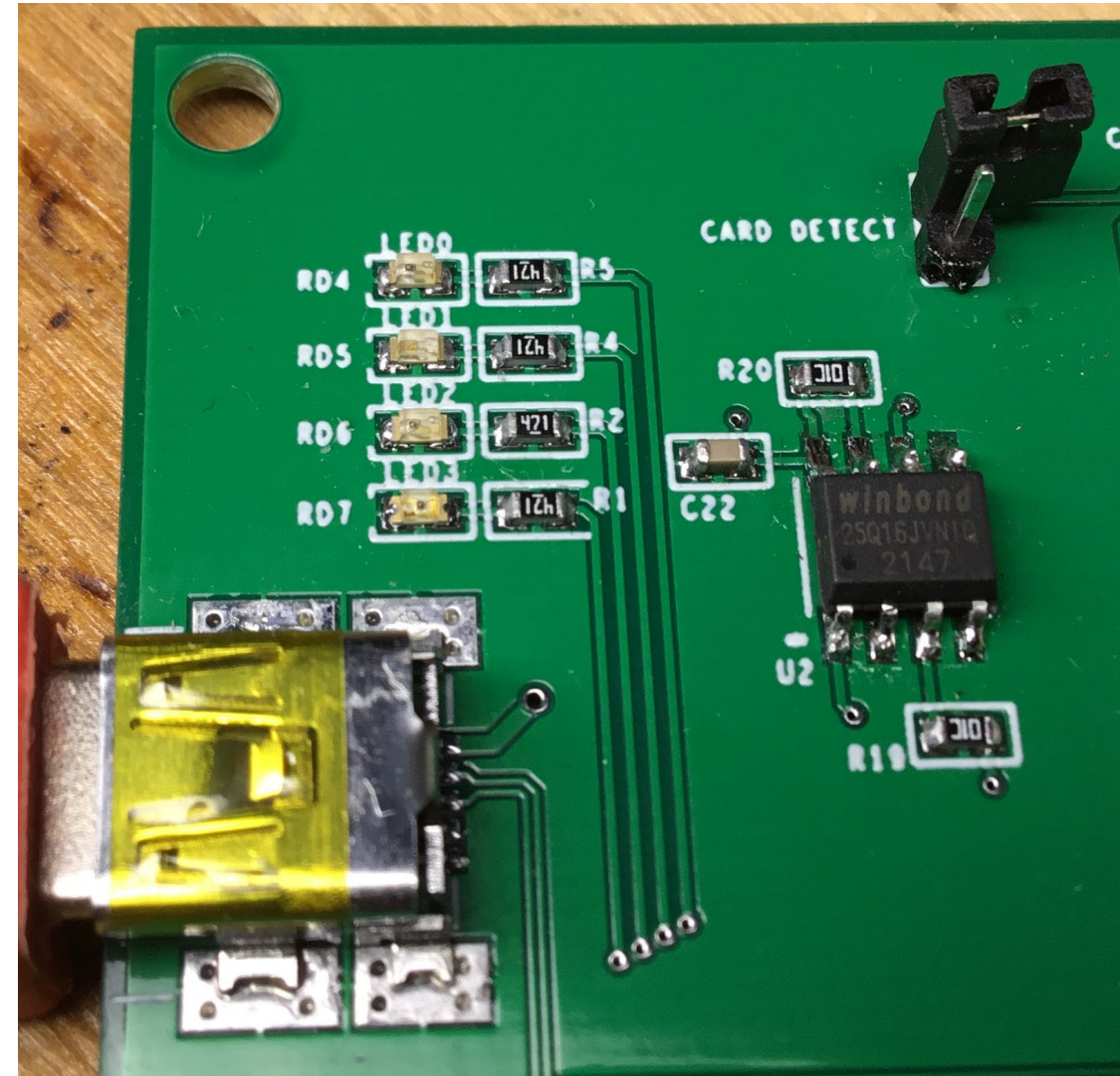
Initialization Code

```
#include <main.h>

unsigned int8 cmdBuf[8];
int16 adcVal;
int1 adcDone;
unsigned int8 cmdBufIndx;

/* TODO: Use usb_cdc_putc() to transmit data to the USB
virtual COM port. Use usb_cdc_kbhit() and usb_cdc_getc() to
receive data from the USB virtual COM port. usb_enumerated()
can be used to see if connected to a host and ready to
communicate. */

void main()
{
    setup_adc_ports(sAN1 | sAN0, VREF_VREF);
    setup_adc(ADC_CLOCK_INTERNAL | ADC_TAD_MUL_8);
    set_adc_channel(0);
    usb_init();
    output_low(led0_ORG);
    output_low(led1_BLU);
    output_low(led2_RED);
    output_low(led3_GRN);
    cmdBufIndx = 0;
}
```



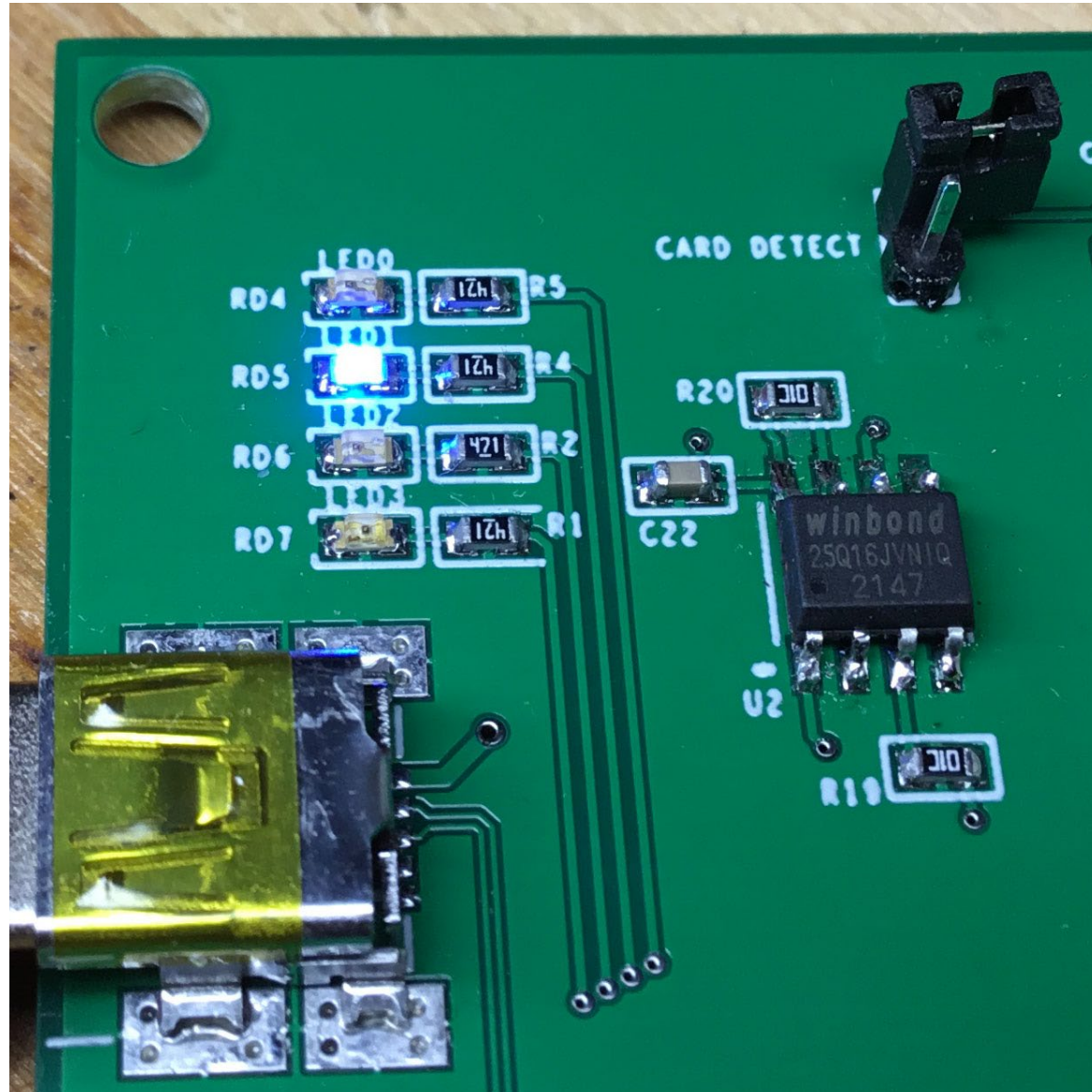
USB Control Application Code

```

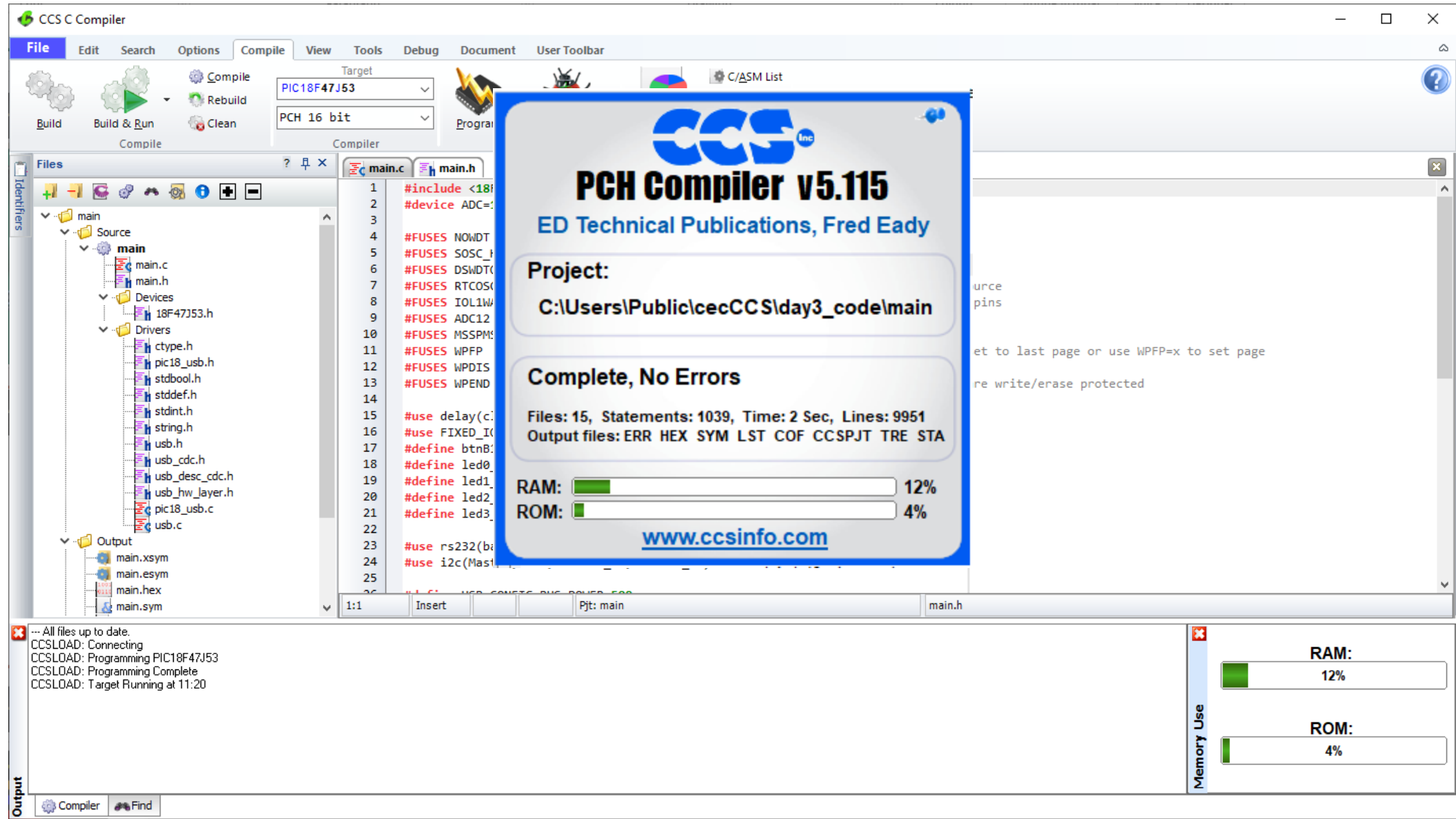
while(TRUE)
{
    if(usb_enumerated())
    {
        if(usb_cdc_kbhit())
        {
            do{
                cmdBuf[cmdBufIndx++] = usb_cdc_getc();
           }while(usb_cdc_kbhit());

            switch(cmdBuf[4])
            {
                case 0xD4:
                    output_high(led0_ORG);
                    output_low(led1_BLU);
                    output_low(led2_RED);
                    output_low(led3_GRN);
                    cmdBufIndx = 0;
                    break;
                case 0xD5:
                    output_low(led0_ORG);
                    output_high(led1_BLU);
                    output_low(led2_RED);
                    output_low(led3_GRN);
                    cmdBufIndx = 0;
                    break;
                case 0xD6:
                    output_low(led0_ORG);
                    output_low(led1_BLU);
                    output_high(led2_RED);
                    output_low(led3_GRN);
                    cmdBufIndx = 0;
                    break;
                case 0xD7:
                    output_low(led0_ORG);
                    output_low(led1_BLU);
                    output_low(led2_RED);
                    output_high(led3_GRN);
                    cmdBufIndx = 0;
                    break;
            }
        }
    }
}

```



Compile the USB Control Application Code



The screenshot shows the CCS C Compiler interface with the following details:

- Compiler Settings:** Target: PIC18F47J53, PCH 16 bit.
- Project Path:** C:\Users\Public\cecCCS\day3_code\main
- Compilation Results:** Complete, No Errors. Files: 15, Statements: 1039, Time: 2 Sec, Lines: 9951. Output files: ERR HEX SYM LST COF CCSPJT TRE STA.
- Memory Usage:** RAM: 12%, ROM: 4%.
- Output Window:**

```

--- All files up to date.
CCSLOAD: Connecting
CCSLOAD: Programming PIC18F47J53
CCSLOAD: Programming Complete
CCSLOAD: Target Running at 11:20

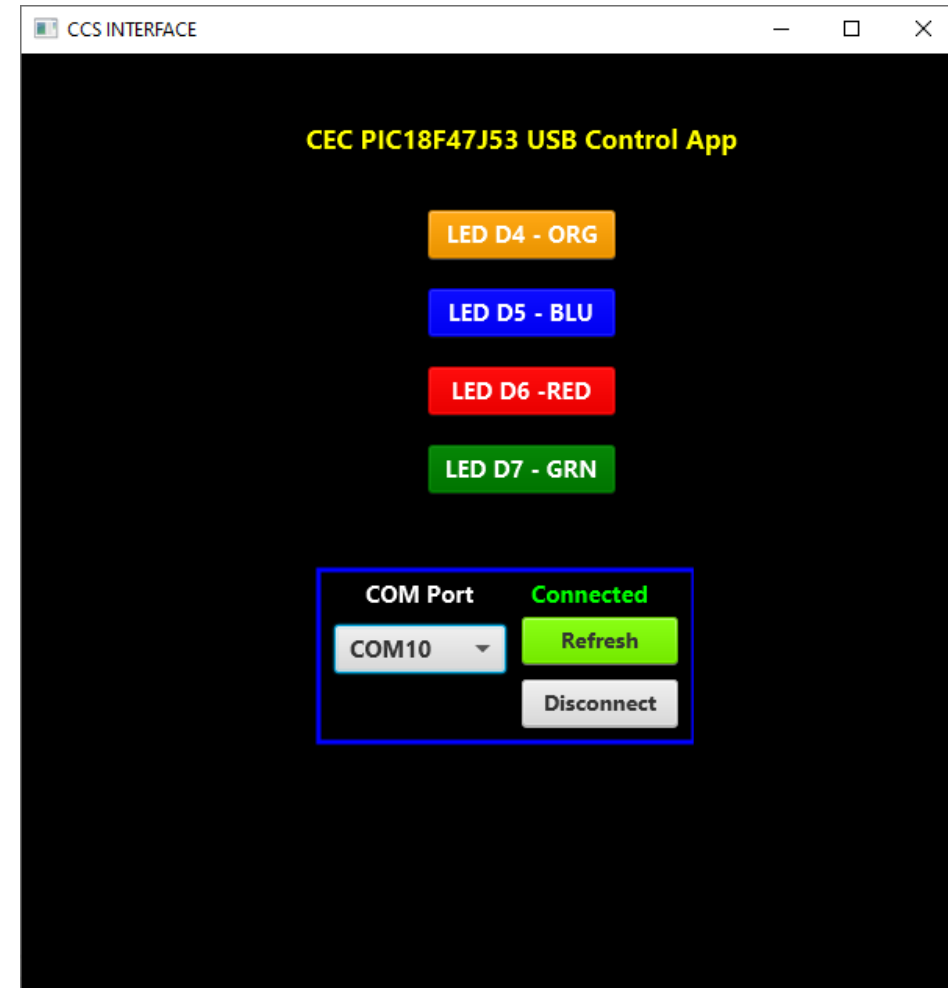
```

USB Interface Application Code

```
Private Sub btnRefreshComPort_Click  
cmbComPorts.Items.Clear  
cmbComPorts.Items.AddAll(usbport.ListPorts)  
End Sub
```

```
Private Sub btnCloseComPort_Click  
Astreams.Write2(cmdBuf,0,2)  
Delay(1000)  
Astreams.Close  
cmbComPorts.Items.Clear  
lblConnectStatus.TextColor = fx.Colors.Red  
lblConnectStatus.Text = "Disconnected"  
End Sub
```

```
Private Sub cmbComPorts_SelectedIndexChanged(Index As Int, Value As Object)  
Log(Index)  
If Index > -1 Then  
usbport.Open(cmbComPorts.Value)  
Astreams.InitializePrefix(usbport.GetInputStream,False,usbport.GetOutputStream,"Astreams")  
lblConnectStatus.TextColor = fx.Colors.Green  
lblConnectStatus.Text = "Connected "  
End If  
End Sub
```



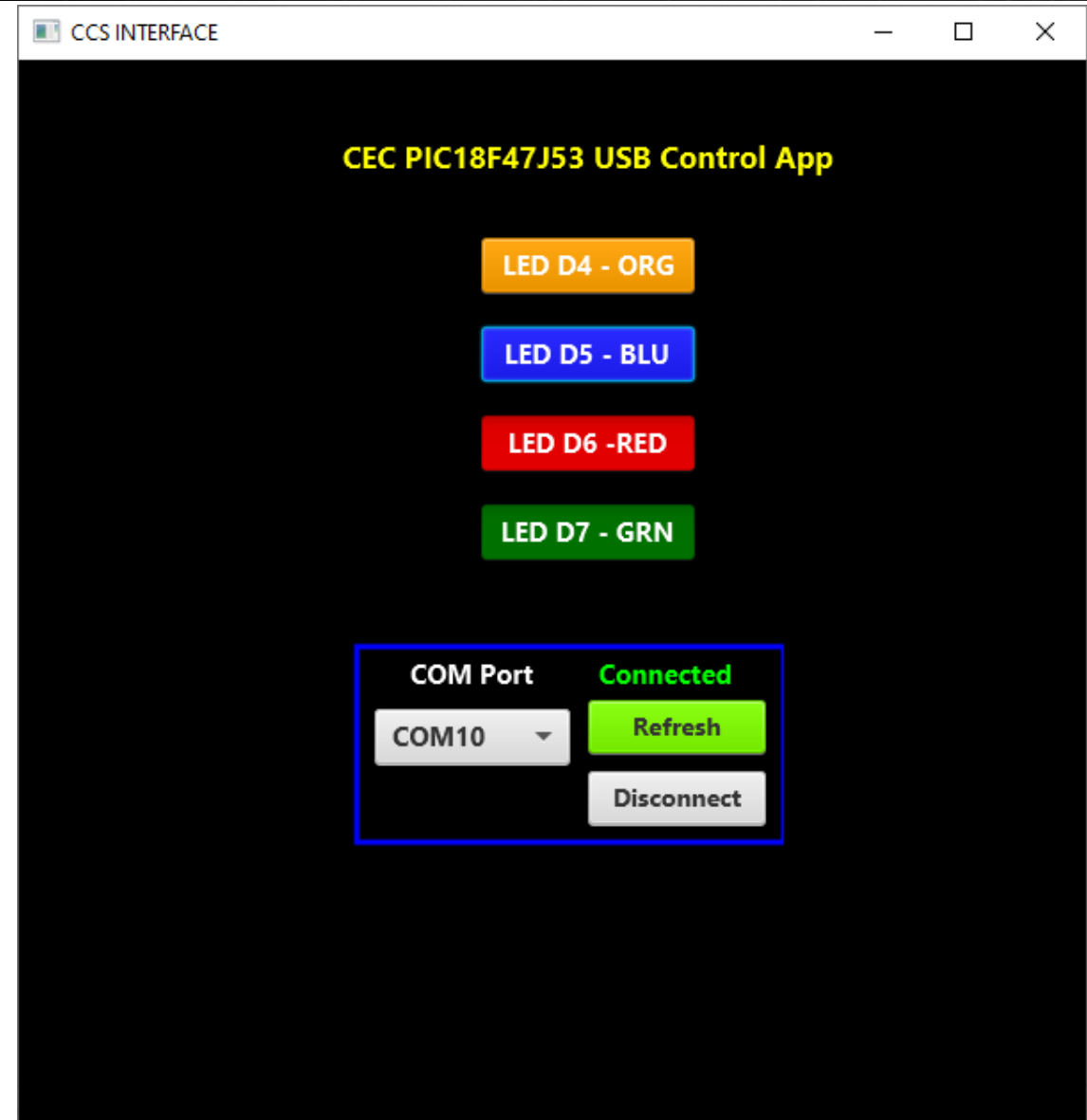
USB Interface Application Code

```
Private Sub togbtnOrg_SelectedChange(Selected As Boolean)
Log("orange")
cmdBuf(0) = 0xD4
Astreams.Write2(cmdBuf,0,1)
End Sub
```

```
Private Sub togbtnBlu_SelectedChange(Selected As Boolean)
Log("blue")
cmdBuf(0) = 0xD5
Astreams.Write2(cmdBuf,0,1)
End Sub
```

```
Private Sub togbtnRed_SelectedChange(Selected As Boolean)
Log("red")
cmdBuf(0) = 0xD6
Astreams.Write2(cmdBuf,0,1)
End Sub
```

```
Private Sub togbtnGrn_SelectedChange(Selected As Boolean)
Log("green")
cmdBuf(0) = 0xD7
Astreams.Write2(cmdBuf,0,1)
End Sub
```



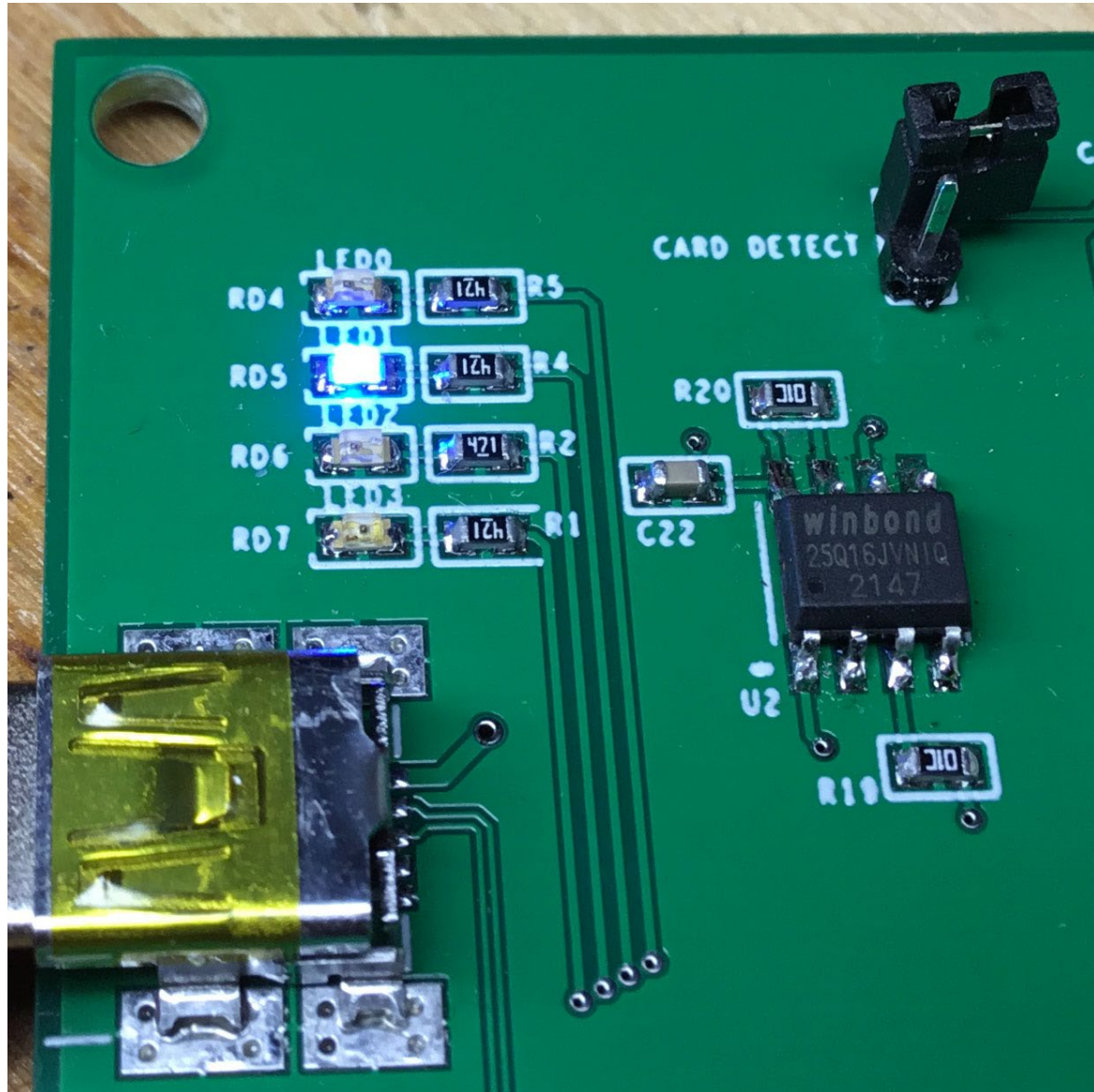
USB Interface Application Code

```
Private Sub togbtnBlu_SelectedChange(Selected As Boolean)  
Log("blue")  
cmdBuf(0) = 0xD5  
Astreams.Write2(cmdBuf,0,1)  
End Sub
```

Expression (in valid C syntax)

cmdBuf [HEX]	
cmdB...	
[0]	0x01
[1]	0x00
[2]	0x00
[3]	0x00
[4]	0xD5
[5]	0x00
[6]	0x8E
[7]	0xDD

USB Interface Application Code



Expression (in valid C syntax)

cmdBuf [HEX]

```

v cmdB...
  ...[0] 0x01
  ...[1] 0x00
  ...[2] 0x00
  ...[3] 0x00
  ...[4] 0xD5
  ...[5] 0x00
  ...[6] 0x8E
  ...[7] 0xDD
  
```

```

if(usb_enumerated())
{
  if(usb_cdc_kbhit())
  {
    do{
      cmdBuf[cmdBufIndx++] = usb_cdc_getc();
    }while(usb_cdc_kbhit());

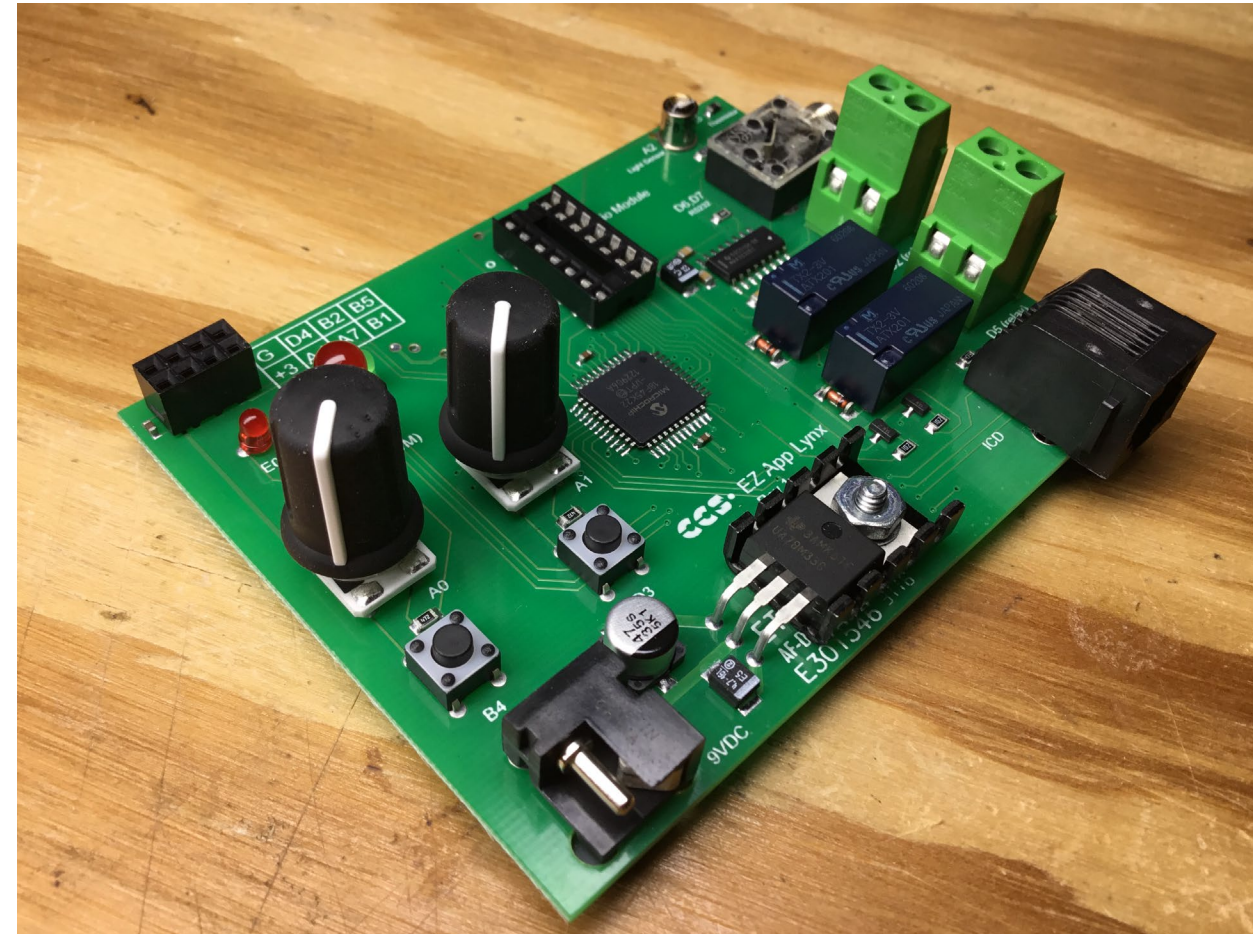
    switch(cmdBuf[4])
    {
      case 0xD4:
        output_high(led0_ORG);
        output_low(led1_BLU);
        output_low(led2_RED);
        output_low(led3_GRN);
        cmdBufIndx = 0;
        break;
      case 0xD5:
        output_low(led0_ORG);
        output_high(led1_BLU);
        output_low(led2_RED);
        output_low(led3_GRN);
        cmdBufIndx = 0;
        break;
    }
  }
}
  
```

MORE TO COME..

Thank you for attending!!!

Please consider the resources below:

- ccsinfo.com
- **CCS C Compiler Manual**
- **Master and Command C for PIC MCU (PDF)**





DesignNews

Thank You

Sponsored by

