# Lab 04 - USB Port I/O

S.THINESH
E/15/366

As the initial step, create a Proteus project with PIC18F4550 as the controller.

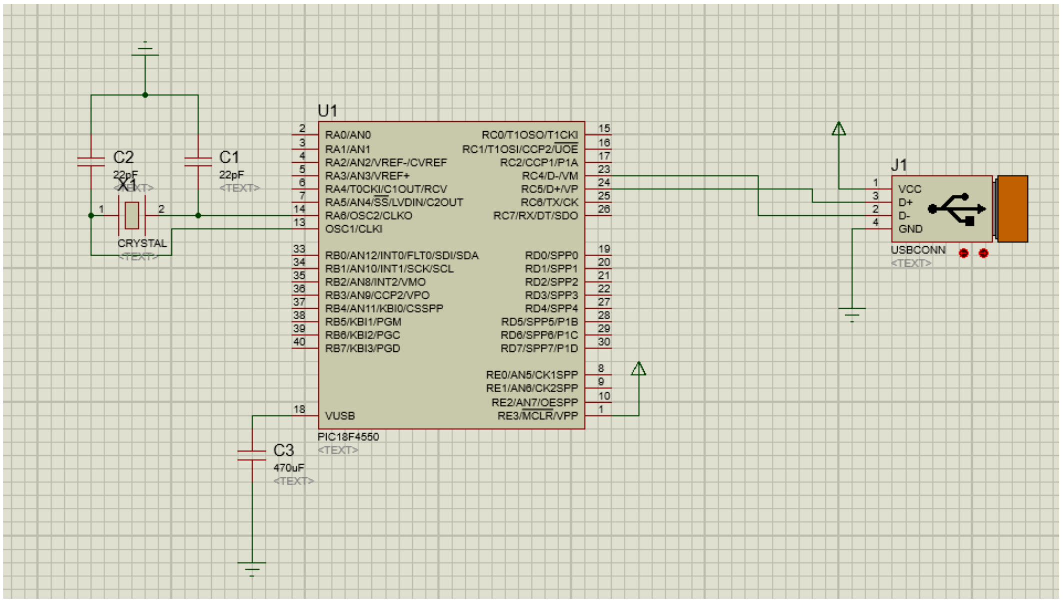
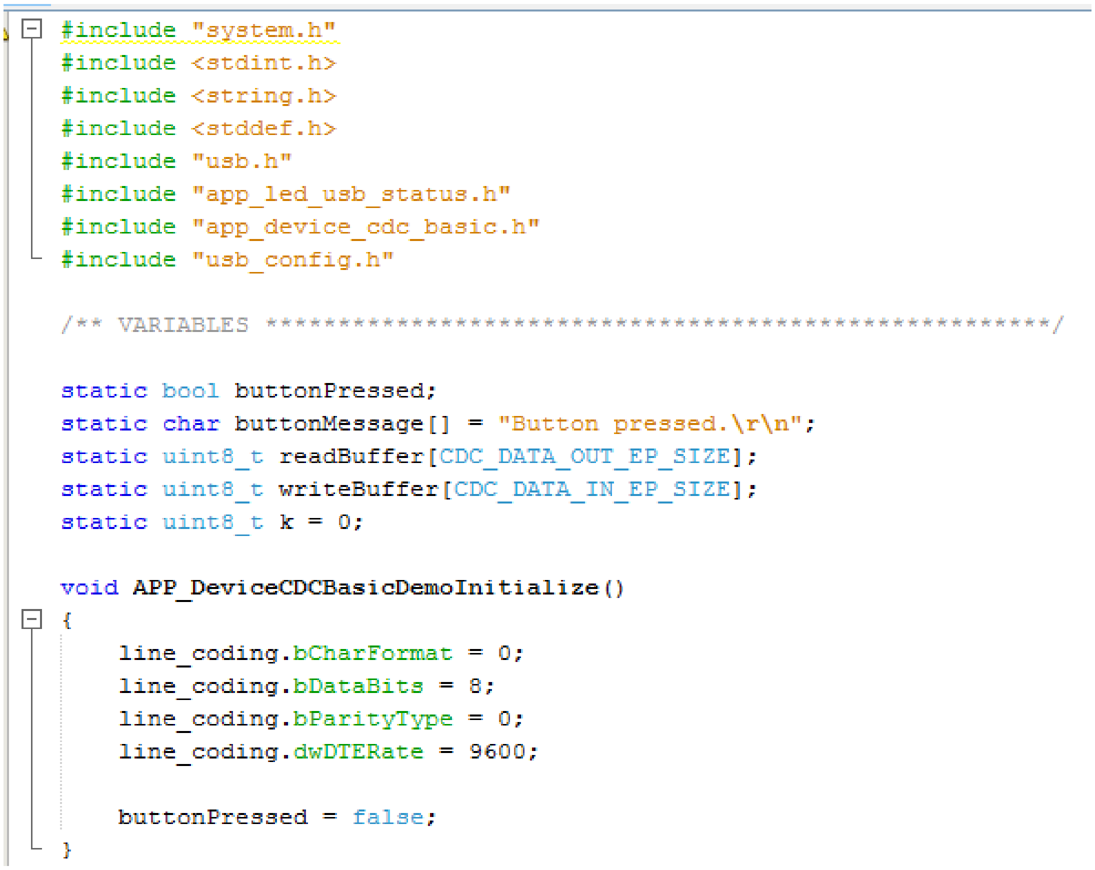
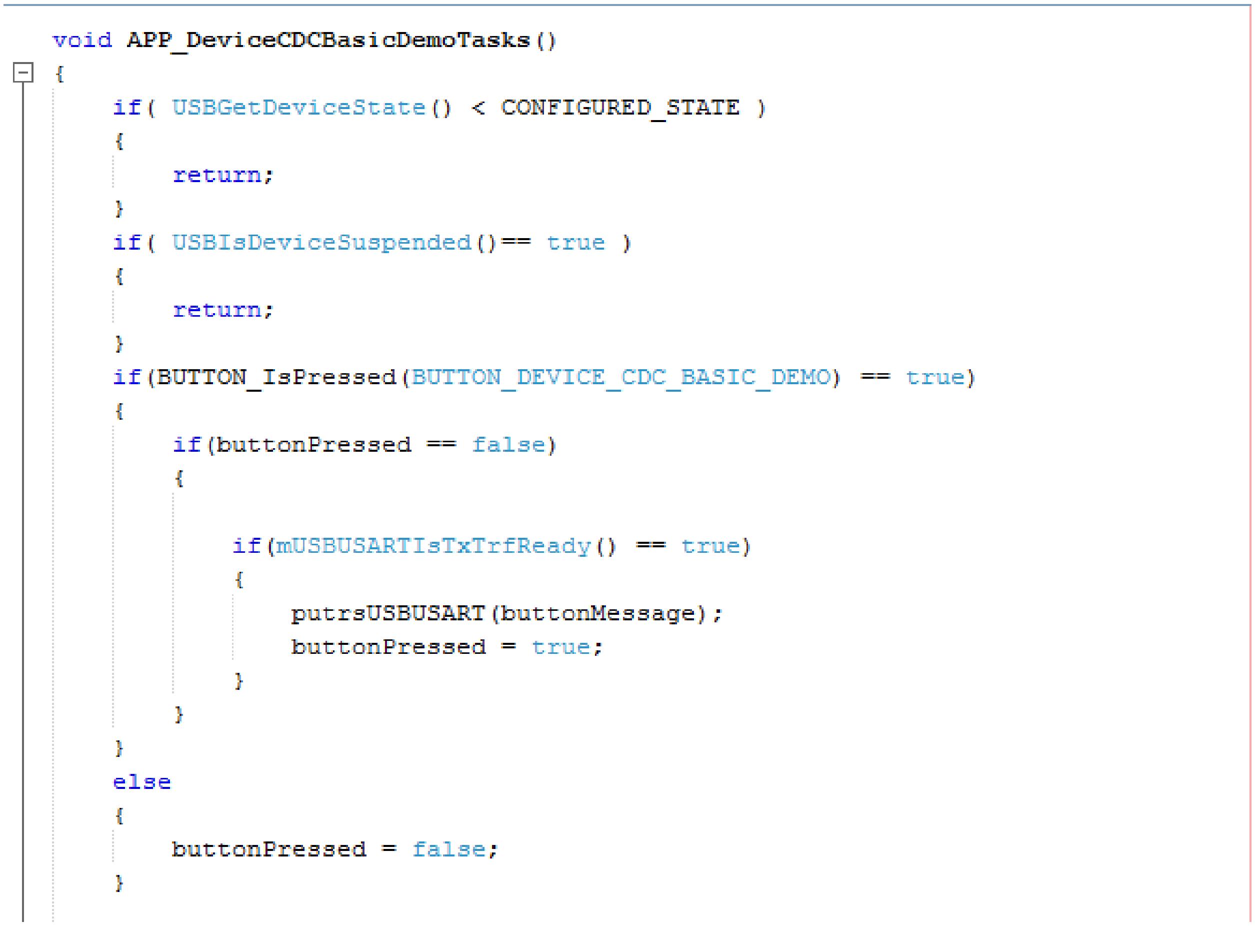


Figure 01

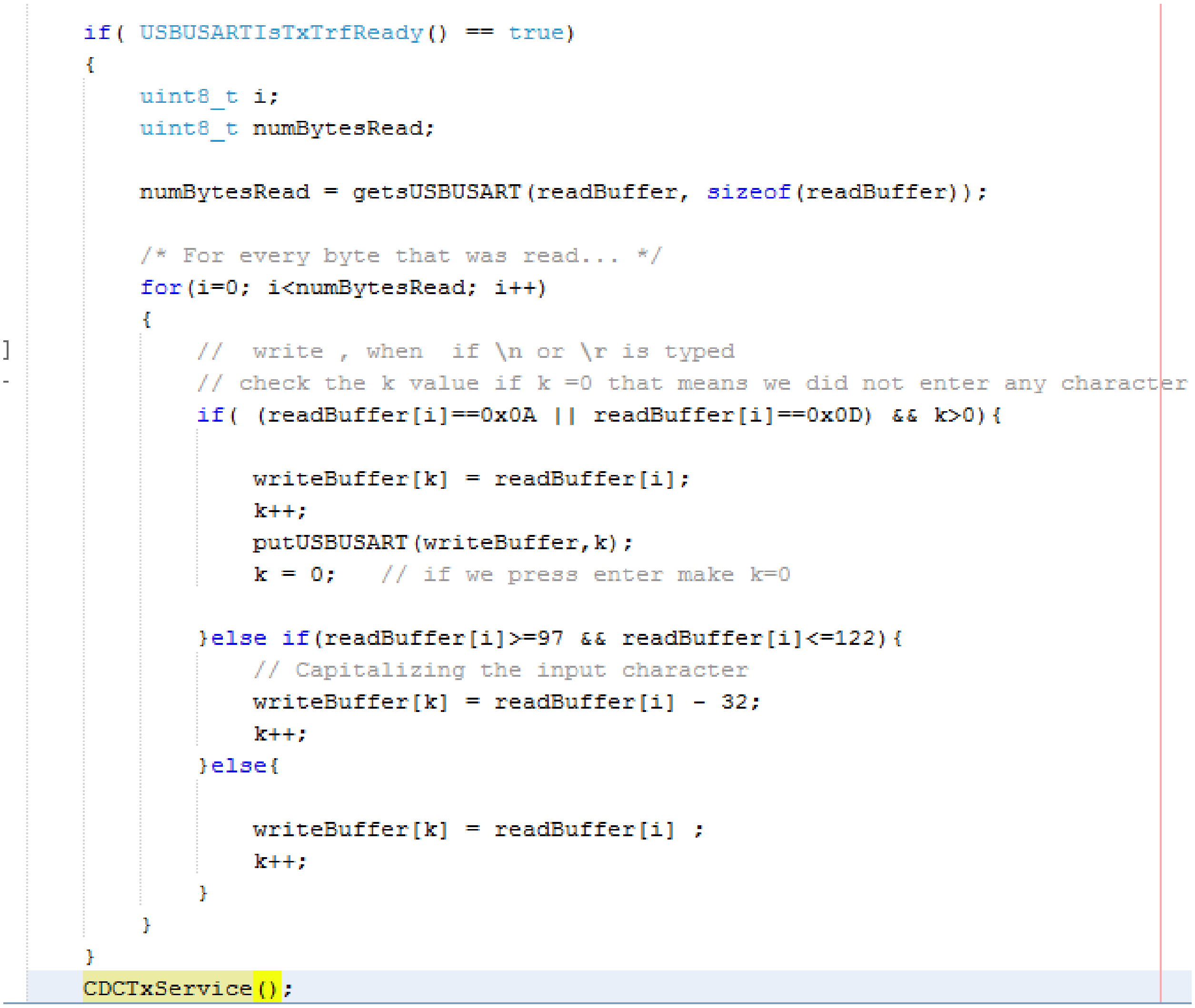
Next use MPLAB X IDE to program the Microcontroller.



```
#include "system.h"
#include <stdint.h>
#include <string.h>
#include <stddef.h>
#include "usb.h"
#include "app_led_usb_status.h"
#include "app_device_cdc_basic.h"
#include "usb_config.h"

/** VARIABLES ******************************************************/

static bool buttonPressed;
static char buttonMessage[] = "Button pressed.\r\n";
static uint8_t readBuffer[CDC_DATA_OUT_EP_SIZE];
static uint8_t writeBuffer[CDC_DATA_IN_EP_SIZE];
static uint8_t k = 0;

void APP_DeviceCDCBasicDemoInitialize()
{
    line_coding.bCharFormat = 0;
    line_coding.bDataBits = 8;
    line_coding.bParityType = 0;
    line_coding.dwDTERate = 9600;

    buttonPressed = false;
}
```

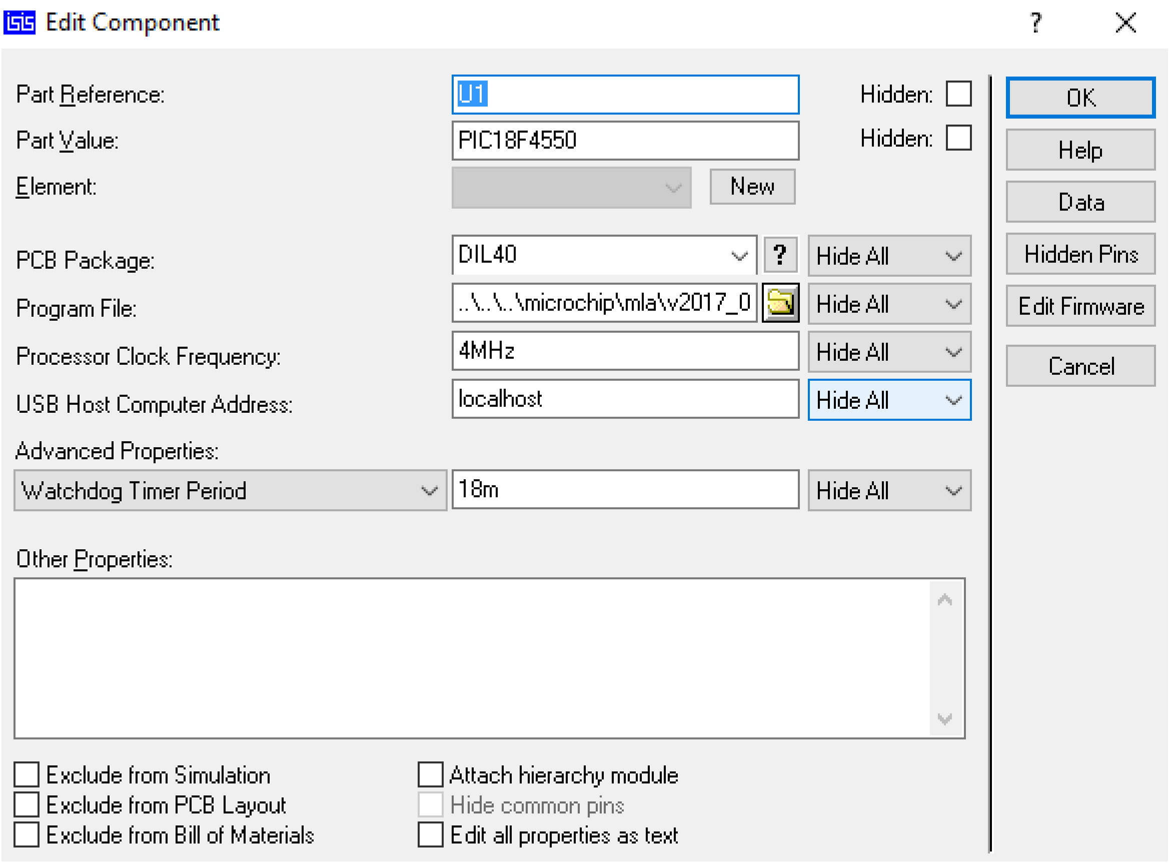


```
void APP_DeviceCDCBasicDemoTasks()
{
    if( USBGetDeviceState() < CONFIGURED_STATE )
    {
        return;
    }
    if( USBIsDeviceSuspended()== true )
    {
        return;
    }
    if(BUTTON_IsPressed(BUTTON_DEVICE_CDC_BASIC_DEMO) == true)
    {
        if(buttonPressed == false)
        {

            if(mUSBUSARTIsTxTrfReady() == true)
            {
                putrsUSBUSART(buttonMessage);
                buttonPressed = true;
            }
        }
    }
    else
    {
        buttonPressed = false;
    }
```



```
    if( USBUSARTIsTxTrfReady() == true)
    {
        uint8_t i;
        uint8_t numBytesRead;

        numBytesRead = getsUSBUSART(readBuffer, sizeof(readBuffer));

        /* For every byte that was read... */
        for(i=0; i<numBytesRead; i++)
        {
            //  write , when  if \n or \r is typed
            // check the k value if k =0 that means we did not enter any character
            if( (readBuffer[i]==0x0A || readBuffer[i]==0x0D) && k>0){

                writeBuffer[k] = readBuffer[i];
                k++;
                putUSBUSART(writeBuffer,k);
                k = 0;   // if we press enter make k=0

            }else if(readBuffer[i]>=97 && readBuffer[i]<=122){
                // Capitalizing the input character
                writeBuffer[k] = readBuffer[i] - 32;
                k++;
            }else{

                writeBuffer[k] = readBuffer[i] ;
                k++;
            }
        }
    }
    CDCTxService();
```

Next step load this generated program code (“.hex file”) to the PIC microcontroller in the Proteus setup. Then run the Proteus simulation.



Edit Component

Part Reference: U1 — Hidden: □

Part Value: PIC18F4550 — Hidden: □

Element: — New

PCB Package: DIL40 — Hide All

Program File: ..\..\..\microchip\mla\v2017_0 — Hide All

Processor Clock Frequency: 4MHz — Hide All

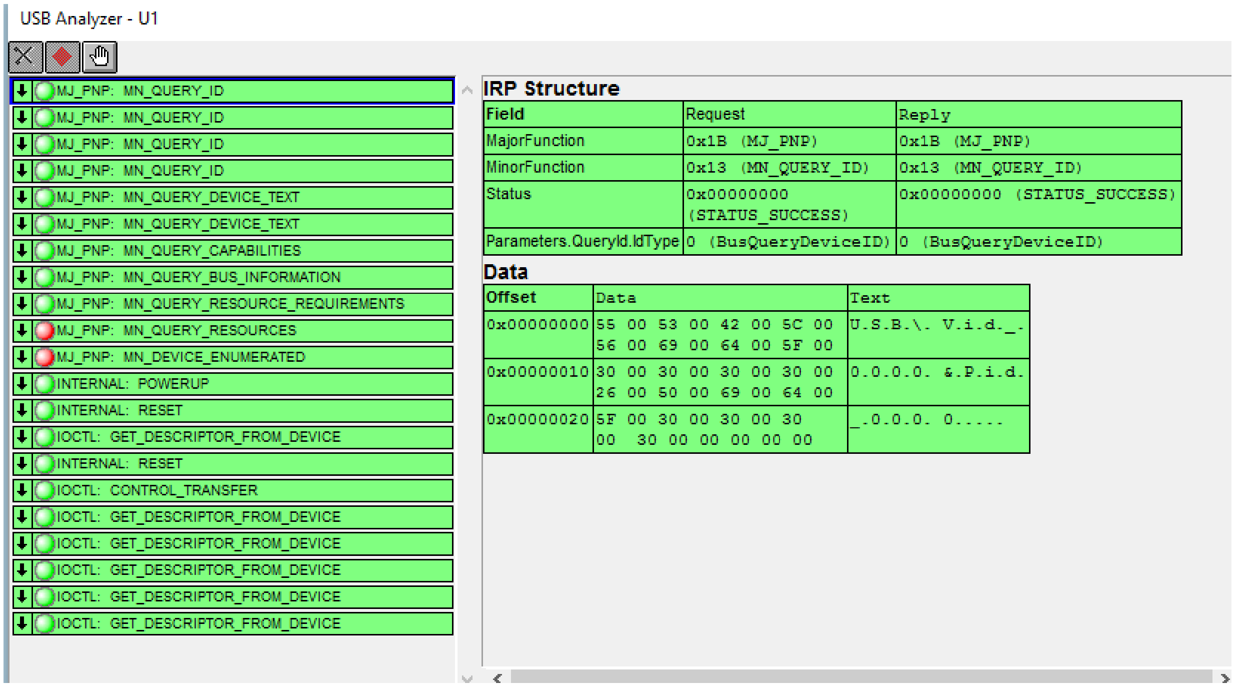
USB Host Computer Address: localhost — Hide All

Advanced Properties:

Watchdog Timer Period: 18m — Hide All

Other Properties:

□ Exclude from Simulation
□ Exclude from PCB Layout
□ Exclude from Bill of Materials
□ Attach hierarchy module
□ Hide common pins
□ Edit all properties as text



USB Analyzer - U1

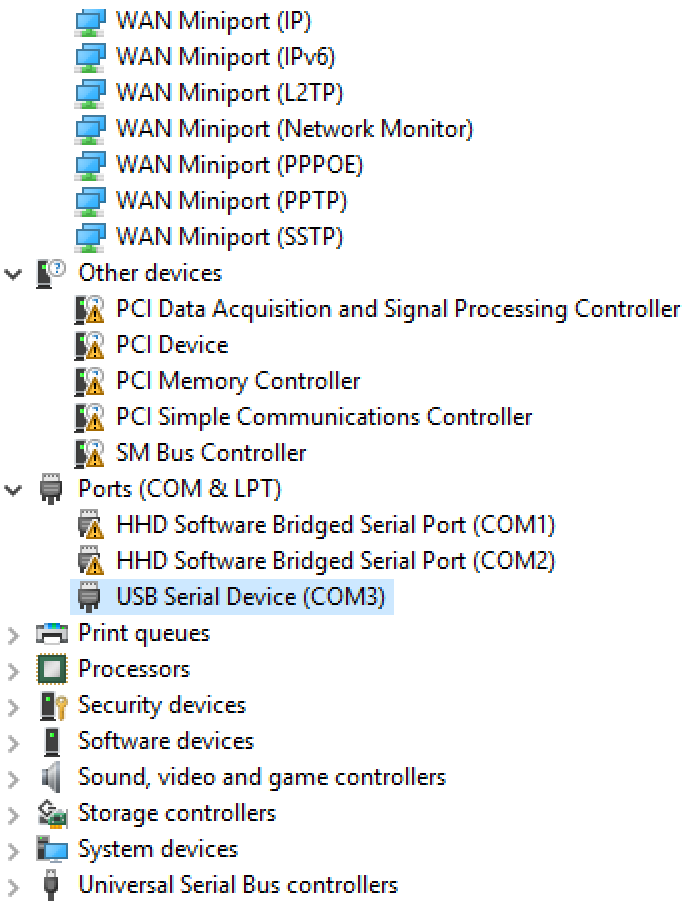
```
MJ_PNP:  MN_QUERY_ID
MJ_PNP:  MN_QUERY_ID
MJ_PNP:  MN_QUERY_ID
MJ_PNP:  MN_QUERY_ID
MJ_PNP:  MN_QUERY_DEVICE_TEXT
MJ_PNP:  MN_QUERY_DEVICE_TEXT
MJ_PNP:  MN_QUERY_CAPABILITIES
MJ_PNP:  MN_QUERY_BUS_INFORMATION
MJ_PNP:  MN_QUERY_RESOURCE_REQUIREMENTS
MJ_PNP:  MN_QUERY_RESOURCES
MJ_PNP:  MN_DEVICE_ENUMERATED
INTERNAL:  POWERUP
INTERNAL:  RESET
IOCTL:  GET_DESCRIPTOR_FROM_DEVICE
INTERNAL:  RESET
IOCTL:  CONTROL_TRANSFER
IOCTL:  GET_DESCRIPTOR_FROM_DEVICE
IOCTL:  GET_DESCRIPTOR_FROM_DEVICE
IOCTL:  GET_DESCRIPTOR_FROM_DEVICE
IOCTL:  GET_DESCRIPTOR_FROM_DEVICE
IOCTL:  GET_DESCRIPTOR_FROM_DEVICE
```

**IRP Structure**

| Field | Request | Reply |
|---|---|---|
| MajorFunction | 0x1B (MJ_PNP) | 0x1B (MJ_PNP) |
| MinorFunction | 0x13 (MN_QUERY_ID) | 0x13 (MN_QUERY_ID) |
| Status | 0x00000000 (STATUS_SUCCESS) | 0x00000000 (STATUS_SUCCESS) |
| Parameters.QueryId.IdType | 0 (BusQueryDeviceID) | 0 (BusQueryDeviceID) |

**Data**

| Offset | Data | Text |
|---|---|---|
| 0x00000000 | 55 00 53 00 42 00 5C 00 56 00 69 00 64 00 5F 00 | U.S.B.\. V.i.d._. |
| 0x00000010 | 30 00 30 00 30 00 30 00 26 00 50 00 69 00 64 00 | 0.0.0.0. &.P.i.d. |
| 0x00000020 | 5F 00 30 00 30 00 30 00 30 00 00 00 00 00 | _.0.0.0. 0..... |



- WAN Miniport (IP)
- WAN Miniport (IPv6)
- WAN Miniport (L2TP)
- WAN Miniport (Network Monitor)
- WAN Miniport (PPPOE)
- WAN Miniport (PPTP)
- WAN Miniport (SSTP)

Other devices

- PCI Data Acquisition and Signal Processing Controller
- PCI Device
- PCI Memory Controller
- PCI Simple Communications Controller
- SM Bus Controller

Ports (COM & LPT)

- HHD Software Bridged Serial Port (COM1)
- HHD Software Bridged Serial Port (COM2)
- USB Serial Device (COM3)

Print queues

Processors

Security devices

Software devices

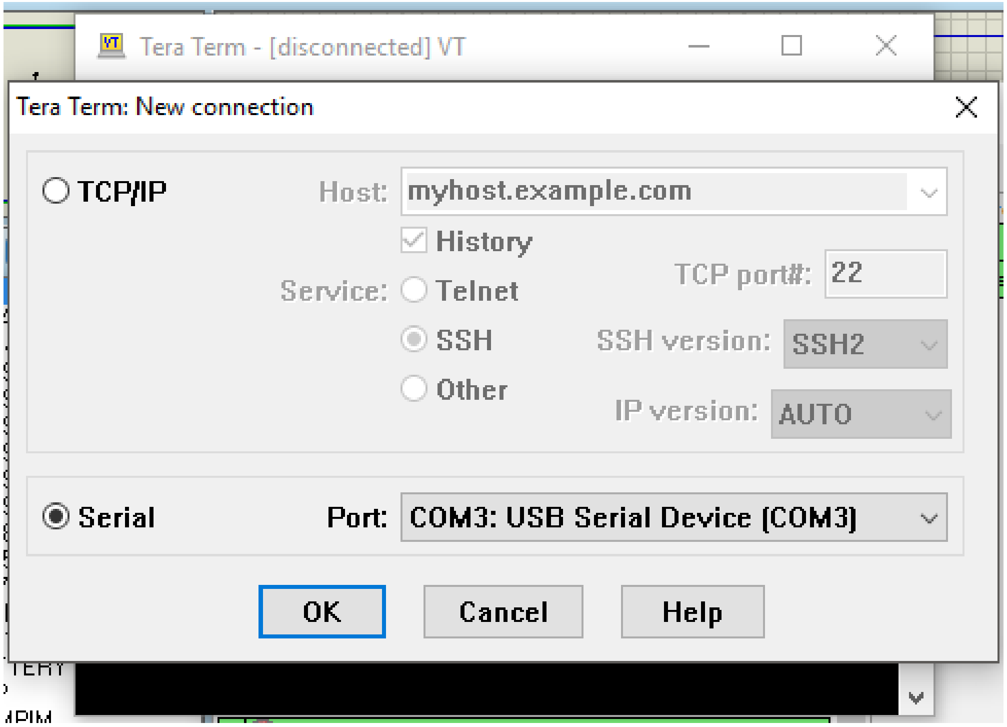
Sound, video and game controllers

Storage controllers

System devices

Universal Serial Bus controllers

Now open TeraTerm. Connect to the corresponding virtual COM port. Type something on the terminal



Tera Term - [disconnected] VT

Tera Term: New connection

○ TCP/IP

Host: myhost.example.com

☑ History

Service: ○ Telnet

● SSH

○ Other

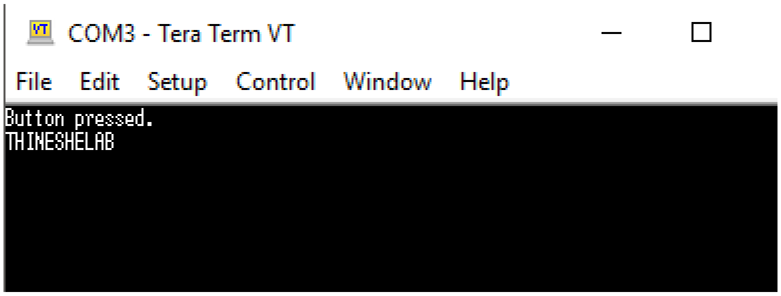
TCP port#: 22

SSH version: SSH2

IP version: AUTO

● Serial

Port: COM3: USB Serial Device (COM3)



```
Button pressed.
THINESHELAB
```

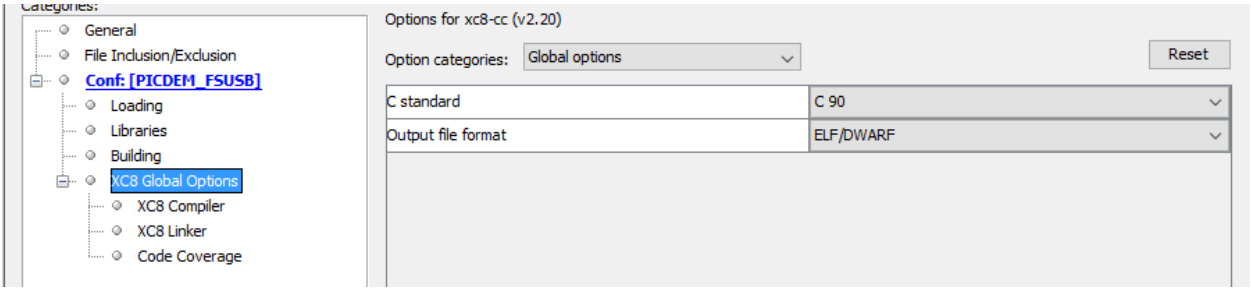
## Problems and issues

- When clean & building the project in MPLAB, I had some errors and building the project was unsuccessful.

In this compilation problem, changing the C standard from 99 to 90 in the compiler and the linker.

Right click on the project -> properties -> XC8 Global Options

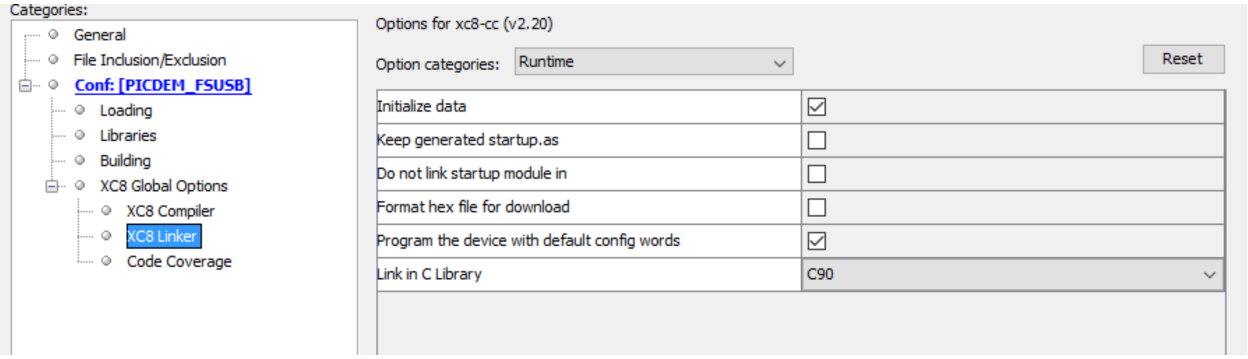
Change C standard from c99 to c90 in CX8 Compiler, CX8 Linker.



Options for xc8-cc (v2.20)

Option categories: Global options

|  |  |
|---|---|
| C standard | C 90 |
| Output file format | ELF/DWARF |

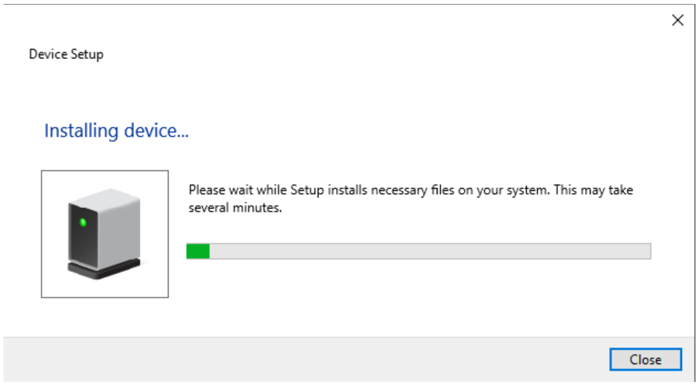


Options for xc8-cc (v2.20)

Option categories: Runtime

|  |  |
|---|---|
| Initialize data | ☑ |
| Keep generated startup.as | □ |
| Do not link startup module in | □ |
| Format hex file for download | □ |
| Program the device with default config words | ☑ |
| Link in C Library | C90 |

- When simulating the project in proteus, new USB device did not recognize by the computer.



Device Setup

Installing device...

Please wait while Setup installs necessary files on your system. This may take several minutes.

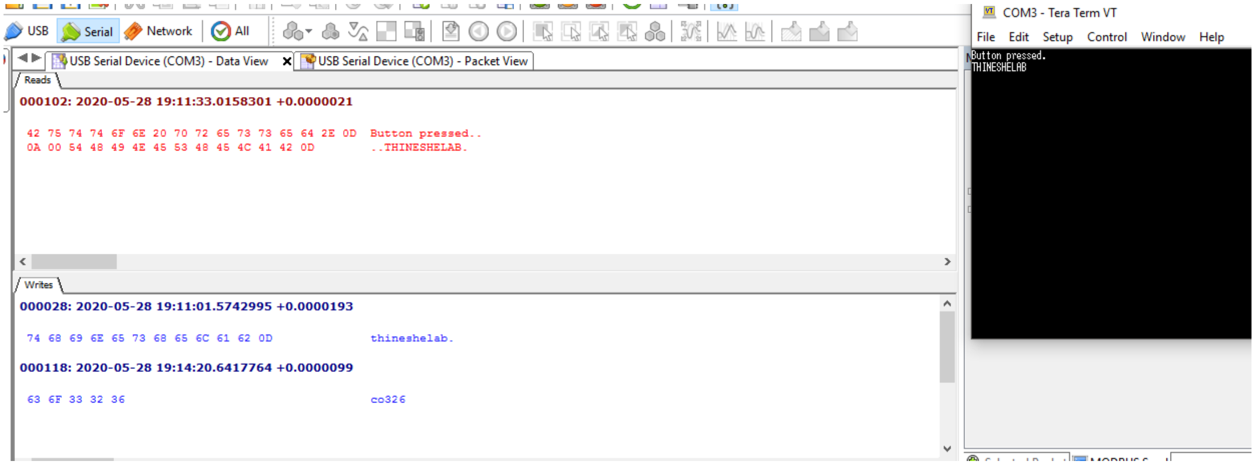
I’ve downloaded and installed the drivers manually to overcome this issue.

1.Give a letter you typed and what is observed on the Tera Term

- when a pressed enter, capitalized letter was displayed in tera term .Otherwise Tera Term Terminal did not display any outputs.

2.Give screenshots of the USB monitor relevant to the letter you type and the letter displayed on the Tera Term.

Letter type

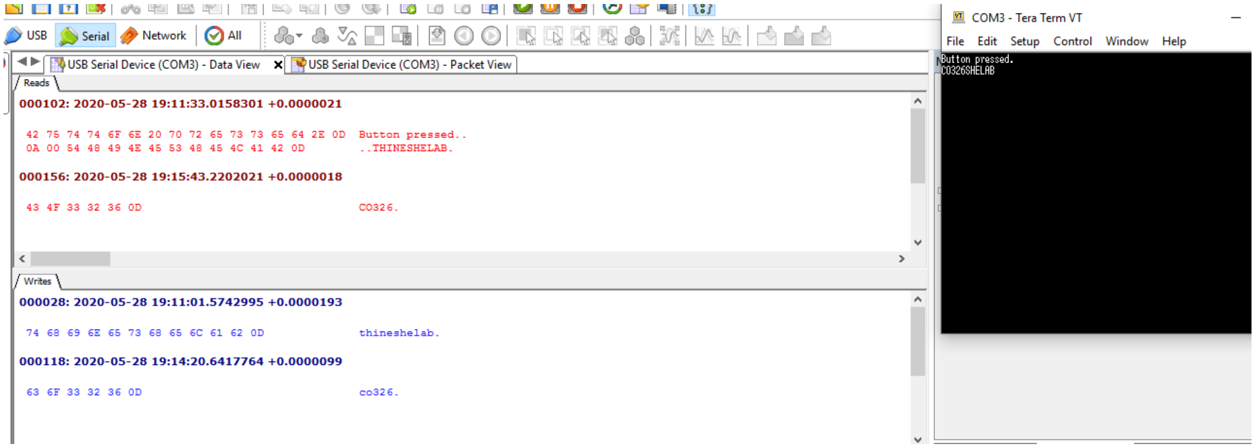


```
000102: 2020-05-28 19:11:33.0158301 +0.0000021

 42 75 74 74 6F 6E 20 70 72 65 73 73 65 64 2E 0D  Button pressed..
 0A 00 54 48 49 4E 45 53 48 45 4C 41 42 0D        ..THINESHELAB.
```

```
000028: 2020-05-28 19:11:01.5742995 +0.0000193

 74 68 69 6E 65 73 68 65 6C 61 62 0D              thineshelab.

000118: 2020-05-28 19:14:20.6417764 +0.0000099

 63 6F 33 32 36                                   co326
```

```
Button pressed.
THINESHELAB
```

Press Enter



```
000102: 2020-05-28 19:11:33.0158301 +0.0000021

 42 75 74 74 6F 6E 20 70 72 65 73 73 65 64 2E 0D  Button pressed..
 0A 00 54 48 49 4E 45 53 48 45 4C 41 42 0D        ..THINESHELAB.

000156: 2020-05-28 19:15:43.2202021 +0.0000018

 43 4F 33 32 36 0D                                CO326.
```

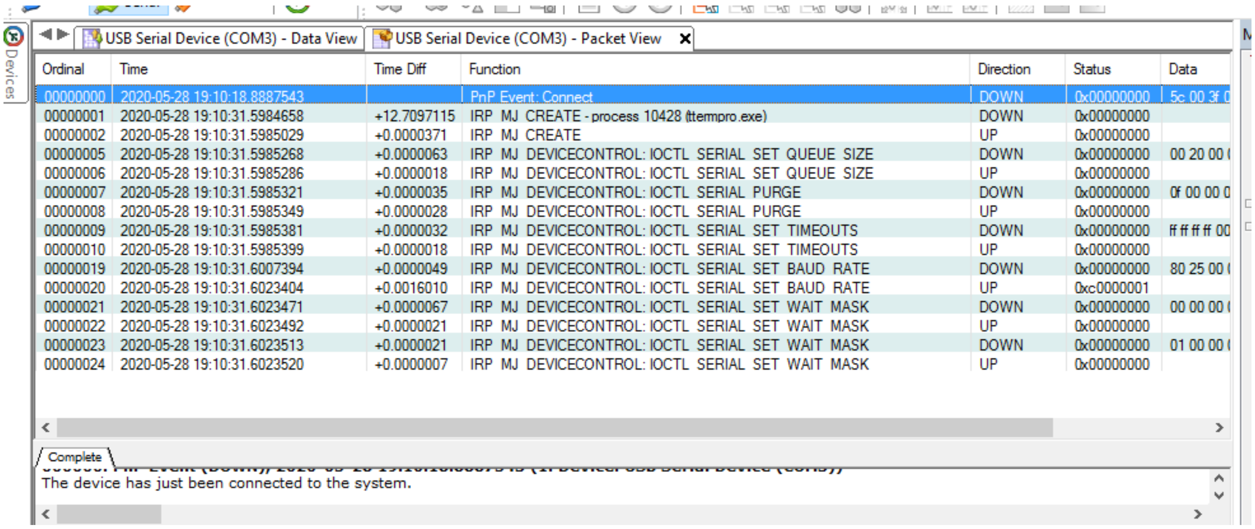
```
000028: 2020-05-28 19:11:01.5742995 +0.0000193

 74 68 69 6E 65 73 68 65 6C 61 62 0D              thineshelab.

000118: 2020-05-28 19:14:20.6417764 +0.0000099

 63 6F 33 32 36 0D                                co326.
```

```
Button pressed.
CO326SHELAB
```

3.One type of packet is IN and other is OUT. Explain each case discussing why they become IN and OUT packets.

Packets sent from Tera Term Terminal to PIC controller are “OUT” packets and packets send from PIC controller to Tera Term Terminal are “IN” packets.

Here Tera Term Terminal is the host and PIC controller is the USB device. When we are typing packets send to PIC, therefore host (TT Terminal) informs USB device (PIC controller) that it wishes to send data, so it’s a “OUT” packet. Similarly, when host reads data from USB device it’s a “IN” packet.



Devices
USB Serial Device (COM3) - Data View
USB Serial Device (COM3) - Packet View
Complete

| Ordinal | Time | Time Diff | Function | Direction | Status | Data |
|---|---|---|---|---|---|---|
| 00000000 | 2020-05-28 19:10:18.8887543 |  | PnP Event: Connect | DOWN | 0x00000000 | 5c 00 3f 0 |
| 00000001 | 2020-05-28 19:10:31.5984658 | +12.7097115 | IRP MJ CREATE - process 10428 (ttermpro.exe) | DOWN | 0x00000000 |  |
| 00000002 | 2020-05-28 19:10:31.5985029 | +0.0000371 | IRP MJ CREATE | UP | 0x00000000 |  |
| 00000005 | 2020-05-28 19:10:31.5985268 | +0.0000063 | IRP MJ DEVICECONTROL: IOCTL SERIAL SET QUEUE SIZE | DOWN | 0x00000000 | 00 20 00 ( |
| 00000006 | 2020-05-28 19:10:31.5985286 | +0.0000018 | IRP MJ DEVICECONTROL: IOCTL SERIAL SET QUEUE SIZE | UP | 0x00000000 |  |
| 00000007 | 2020-05-28 19:10:31.5985321 | +0.0000035 | IRP MJ DEVICECONTROL: IOCTL SERIAL PURGE | DOWN | 0x00000000 | 0f 00 00 0 |
| 00000008 | 2020-05-28 19:10:31.5985349 | +0.0000028 | IRP MJ DEVICECONTROL: IOCTL SERIAL PURGE | UP | 0x00000000 |  |
| 00000009 | 2020-05-28 19:10:31.5985381 | +0.0000032 | IRP MJ DEVICECONTROL: IOCTL SERIAL SET TIMEOUTS | DOWN | 0x00000000 | ff ff ff ff 00 |
| 00000010 | 2020-05-28 19:10:31.5985399 | +0.0000018 | IRP MJ DEVICECONTROL: IOCTL SERIAL SET TIMEOUTS | UP | 0x00000000 |  |
| 00000019 | 2020-05-28 19:10:31.6007394 | +0.0000049 | IRP MJ DEVICECONTROL: IOCTL SERIAL SET BAUD RATE | DOWN | 0x00000000 | 80 25 00 ( |
| 00000020 | 2020-05-28 19:10:31.6023404 | +0.0016010 | IRP MJ DEVICECONTROL: IOCTL SERIAL SET BAUD RATE | UP | 0xc0000001 |  |
| 00000021 | 2020-05-28 19:10:31.6023471 | +0.0000067 | IRP MJ DEVICECONTROL: IOCTL SERIAL SET WAIT MASK | DOWN | 0x00000000 | 00 00 00 ( |
| 00000022 | 2020-05-28 19:10:31.6023492 | +0.0000021 | IRP MJ DEVICECONTROL: IOCTL SERIAL SET WAIT MASK | UP | 0x00000000 |  |
| 00000023 | 2020-05-28 19:10:31.6023513 | +0.0000021 | IRP MJ DEVICECONTROL: IOCTL SERIAL SET WAIT MASK | DOWN | 0x00000000 | 01 00 00 ( |
| 00000024 | 2020-05-28 19:10:31.6023520 | +0.0000007 | IRP MJ DEVICECONTROL: IOCTL SERIAL SET WAIT MASK | UP | 0x00000000 |  |

```
The device has just been connected to the system.
```