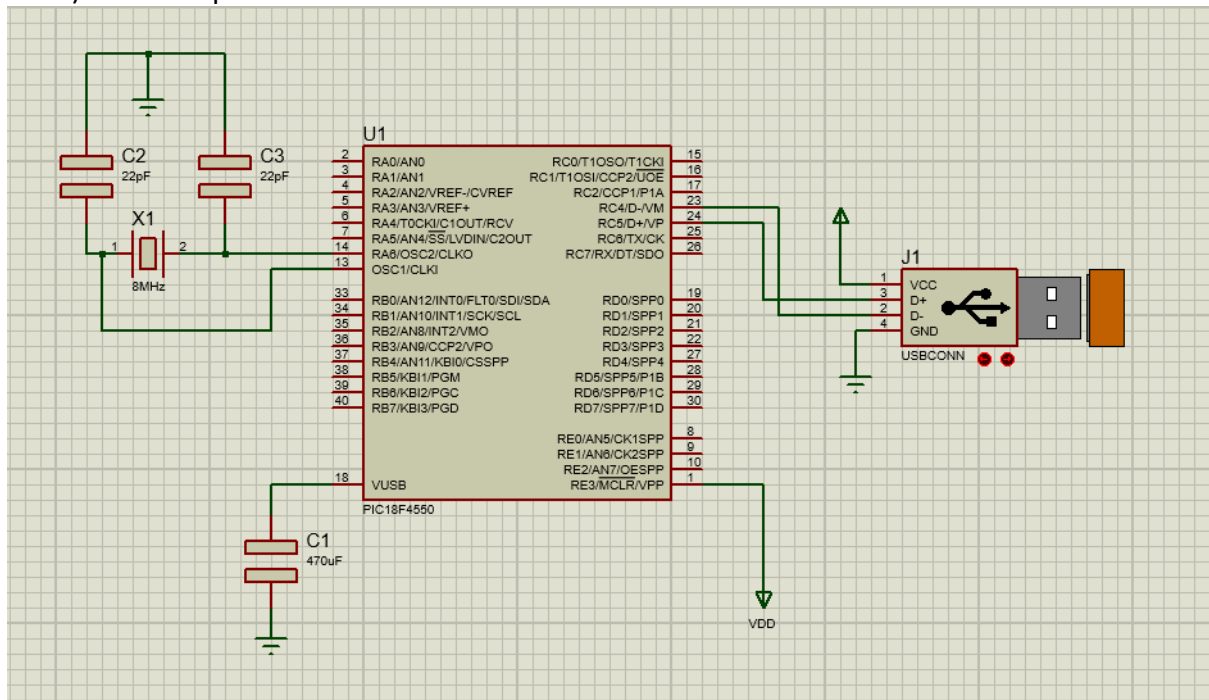


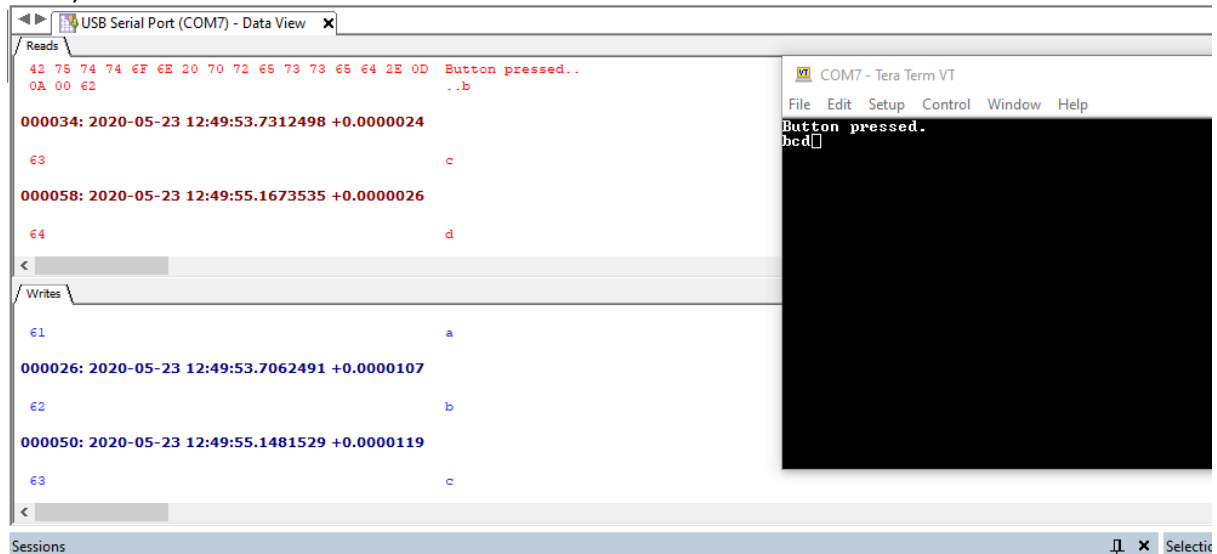
Lab 04 – USB Port I/O

E/15/202

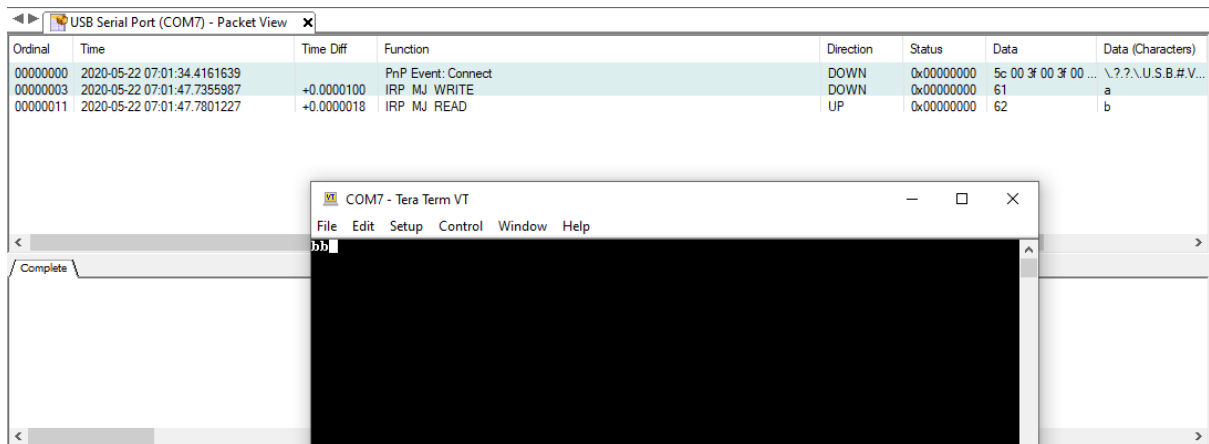
1) Lab setup



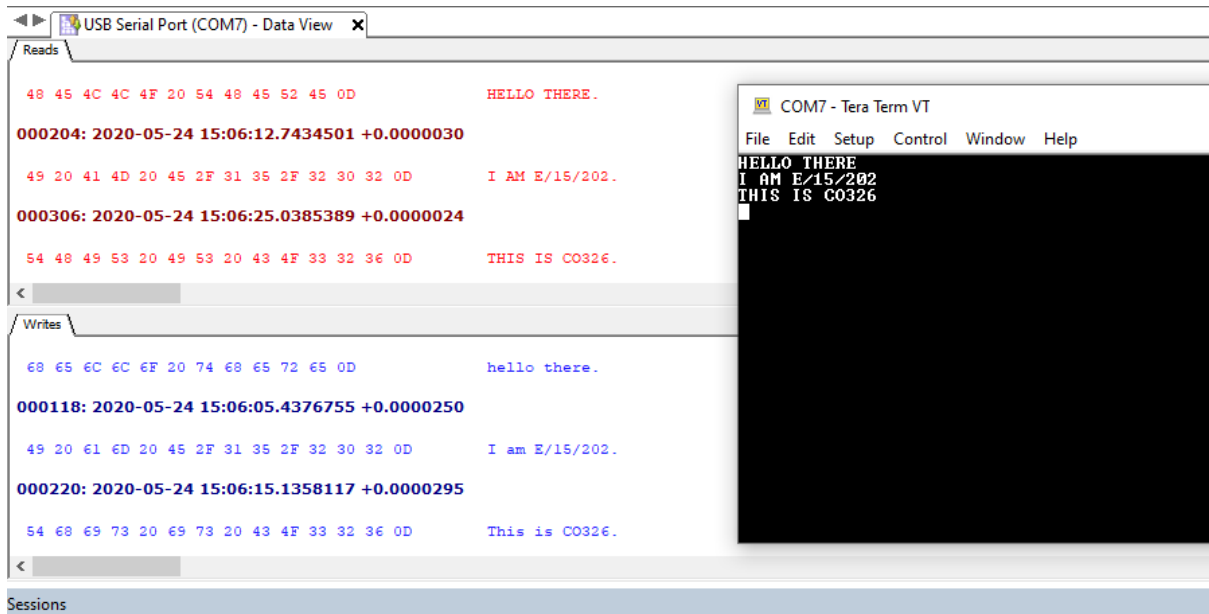
2) Tera terminal with device monitor in Data view



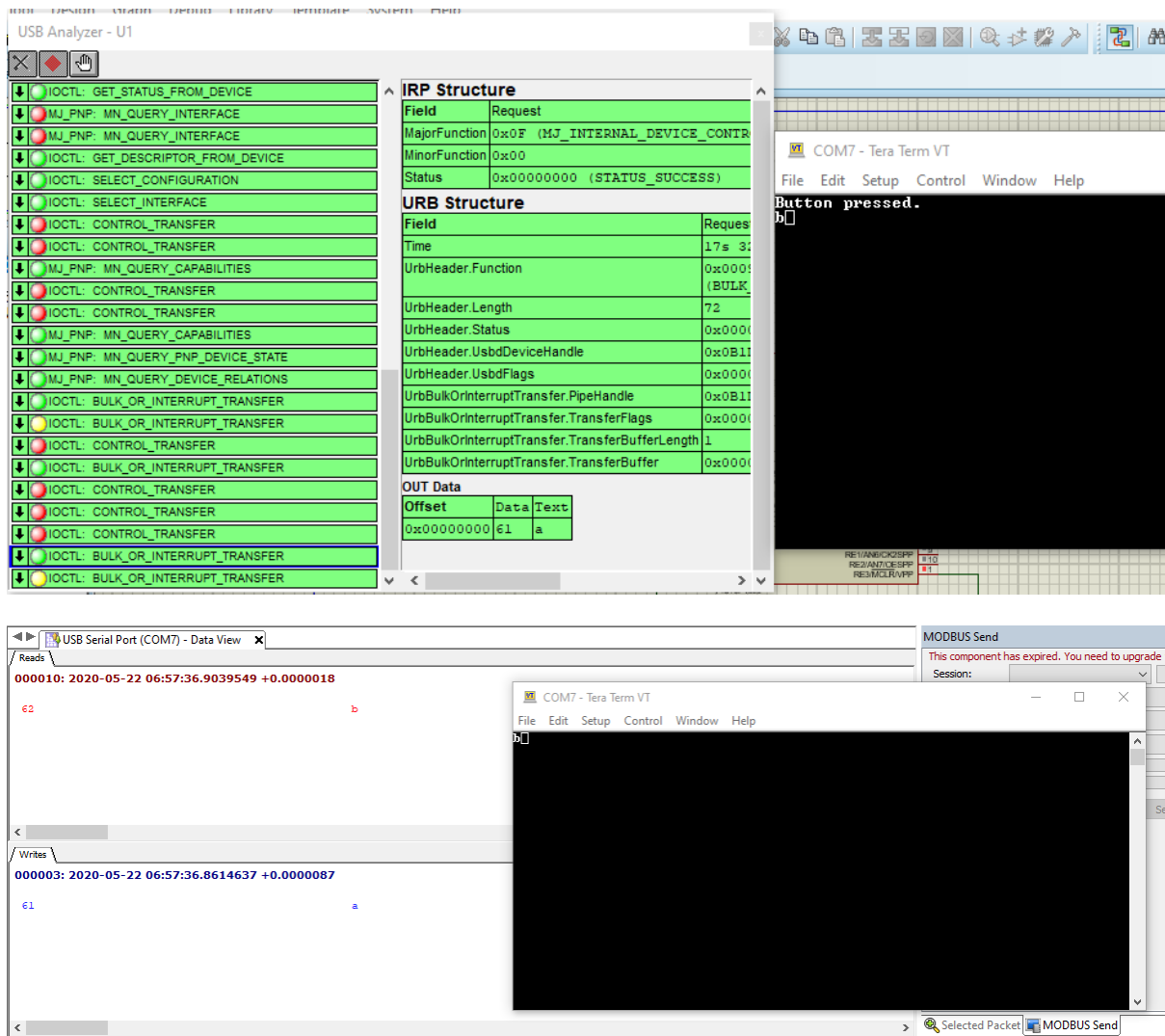
3) Tera terminal with device monitor in packet view



- 4) app_device_cdc_basic.c (lab task code without while loop – but works repeatedly)
 app_new.c (lab task code with a while loop – no change)
 Screenshot of tera term with device monitor for the above codes



To get the above had to change the tera term terminal setup



6) Packet OUT or OUT data means the data written by the user to the terminal. This becomes OUT data because it is the data that sent to the PIC microcontroller. It leaves the USB port.

Packet IN or IN data means the processed data that is printed on the terminal, this becomes IN data because it is an incoming packet from the PIC microcontroller to the USB port.

(Although USB Analyzer shows two types as OUT and IN data, for the first input it shows only the OUT data transfer packet details (see the screenshots of (5)). But from that onwards it shows both IN and OUT data transfer packet details.)

USB Analyzer - U1

IRP Structure

| Field | Request |
|---------------|-----------------------------------|
| MajorFunction | 0x0F (MJ_INTERNAL_DEVICE_CONTROL) |
| MinorFunction | 0x00 |
| Status | 0x00000000 (STATUS_SUCCESS) |

URB Structure

| Field | Request |
|---|-------------------------------------|
| Time | 33s 87 |
| UrbHeader.Function | 0x0000 (BULK_OR_INTERRUPT_TRANSFER) |
| UrbHeader.Length | 72 |
| UrbHeader.Status | 0x0000 |
| UrbHeader.UsbdDeviceHandle | 0x1AE |
| UrbHeader.UsbdFlags | 0x0000 |
| UrbBulkOrInterruptTransfer.PipeHandle | 0x1AE |
| UrbBulkOrInterruptTransfer.TransferFlags | 0x0000 |
| UrbBulkOrInterruptTransfer.TransferBufferLength | 1 |
| UrbBulkOrInterruptTransfer.TransferBuffer | 0x0000 |

OUT Data

| Offset | Data | Text |
|------------|------|------|
| 0x00000000 | 62 | b |

USB Analyzer - U1

IRP Structure

| Field | Request |
|---------------|-----------------------------------|
| MajorFunction | 0x0F (MJ_INTERNAL_DEVICE_CONTROL) |
| MinorFunction | 0x00 |
| Status | 0xC00000BB (STATUS_NOT_SUPPORTED) |

URB Structure

| Field | Request |
|---|-------------------------------------|
| Time | 21s 20 |
| UrbHeader.Function | 0x0000 (BULK_OR_INTERRUPT_TRANSFER) |
| UrbHeader.Length | 72 |
| UrbHeader.Status | 0x0000 |
| UrbHeader.UsbdDeviceHandle | 0x1AE |
| UrbHeader.UsbdFlags | 0x0000 |
| UrbBulkOrInterruptTransfer.PipeHandle | 0x1AE |
| UrbBulkOrInterruptTransfer.TransferFlags | 0x0000 |
| UrbBulkOrInterruptTransfer.TransferBufferLength | 4096 |
| UrbBulkOrInterruptTransfer.TransferBuffer | 0x0000 |

IN Data

| Offset | Data | Text |
|------------|------|------|
| 0x00000000 | 63 | c |

7) Problems and issues with my solutions

- Unable to build the project `picdem_fs_usb.x` in mla version `v2017_03_06`. So I had to uninstall it and install the current mla version `v2018_11_26` for windows.
- When doing the lab task first I had to identify which code that I have to modify in the given project. By going through some of the C codes inside the project was able to find the correct code to modify.
- Understanding about the functions inside the code (`getsUSBUSART()`, `putUSBUSART()`, `putsUSBUSART()`). Had to refer some internet articles and understood how the caesar cipher is done.

- The data types of the inputs are different. Some functions need byte inputs and others in a char array or constant char array. So had to refer the function declarations in their corresponding header file.
- When usb is ready for transmission and then when we use `getsUSBUSART()` function to get the inputs sometimes it reads garbage inputs even though we did not press any key. To identify that device monitor was helpful. To avoid that had to tell it to proceed only if the number of bytes it read is greater than zero.
- If whole code (labtask code) was put under only one *`if(USBUSARTIsTxTrfReady())`* clause then it did not give the expected output. Although the characters were converted to capital when they are printed in the tera terminal they printed in an unexpected way. So I did it in 3 steps. At the first usb ready if clause I only get the whole input from user and converted them to capital and in the second usb ready if clause the converted sentence was sent to the terminal to print and at last included a new line.
- It say lab task should happen in a loop so I included the inside a while (1) loop but it doesn't change anything. Without that loop also the code functions repeatedly.