UART Programming

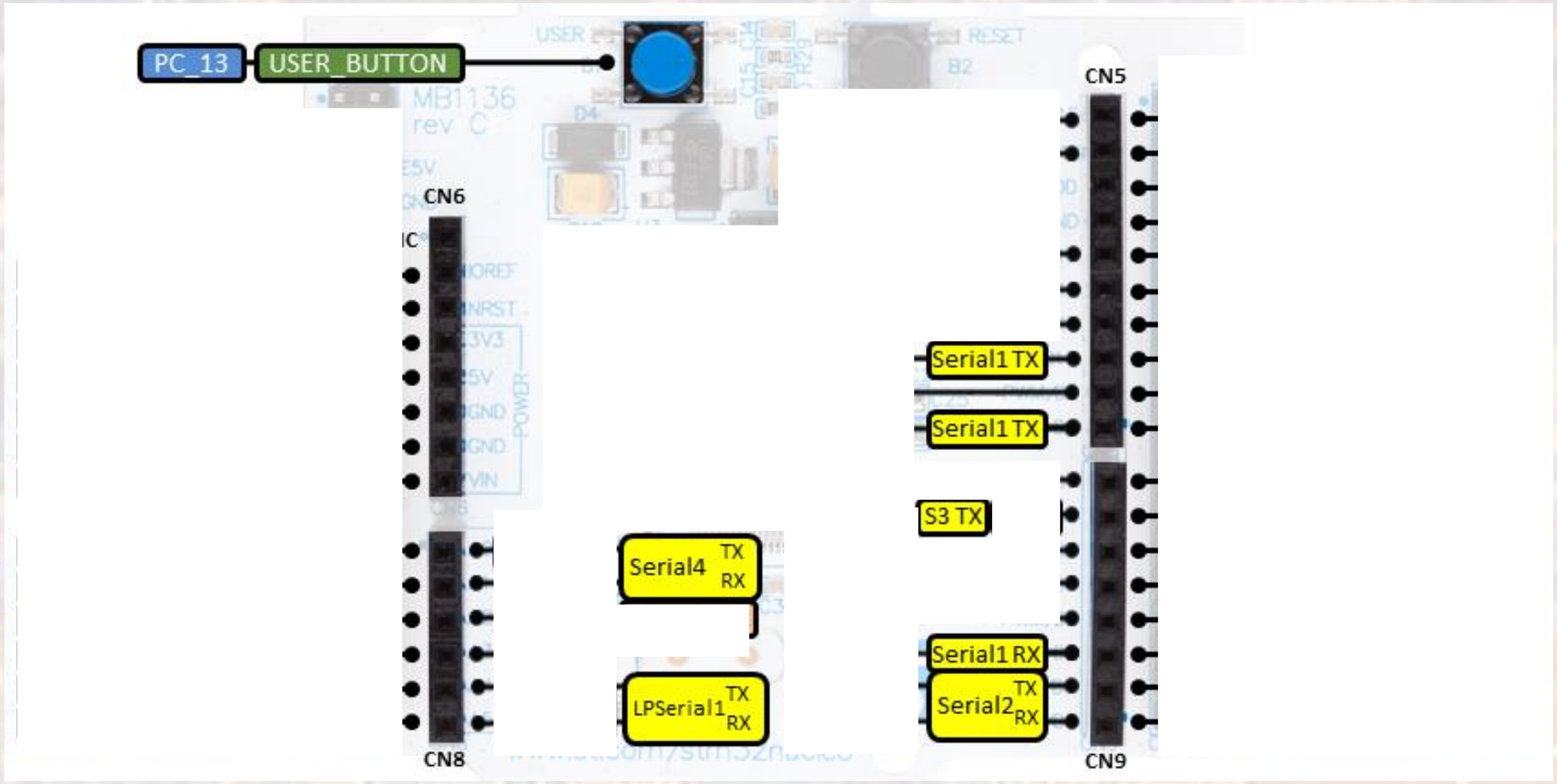
Last updated 6/30/21

UART Programming

- UART Operation
  - Nucleo-L476RG has 5 UART modules
  - 3 available on the Arduino headers
  - 2 additional available on the Morpho headers

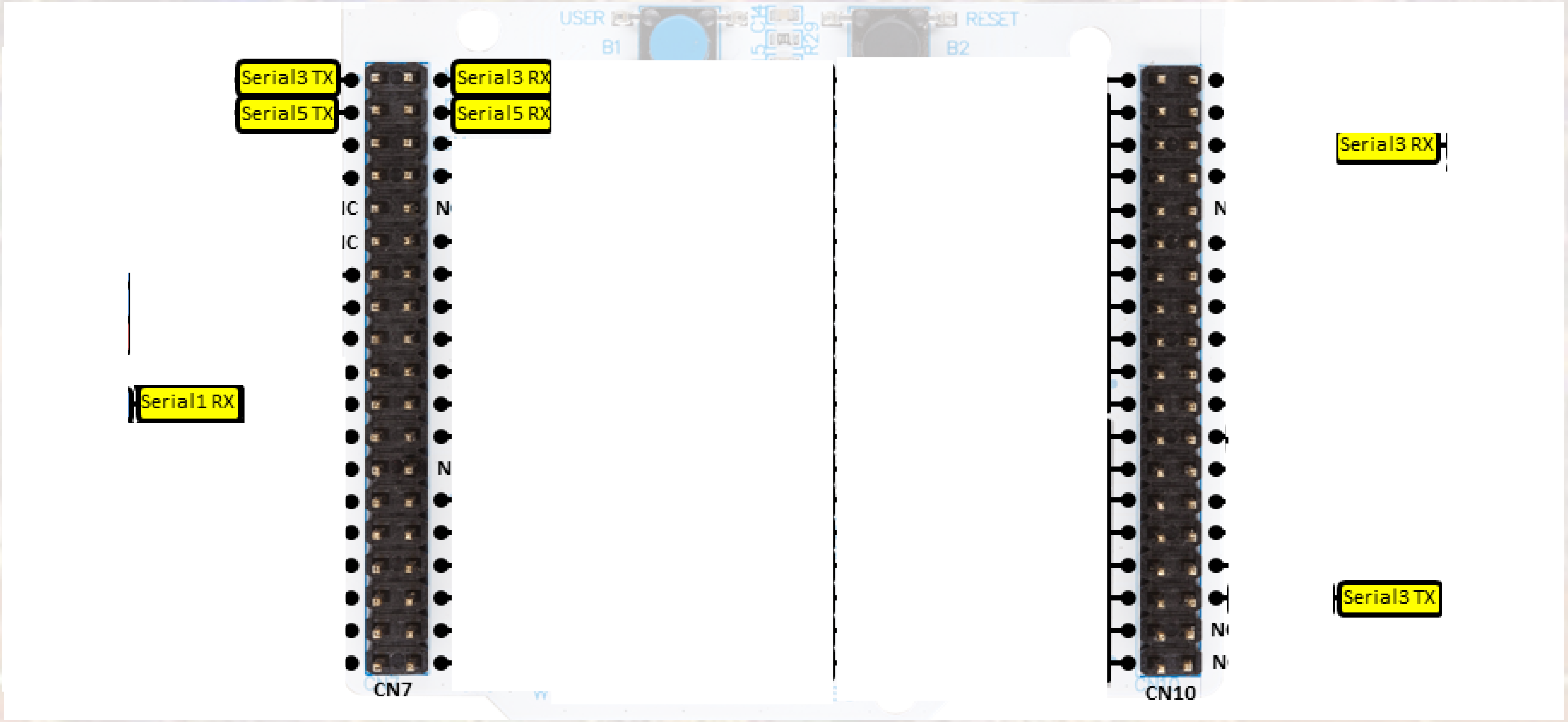
UART Programming

- UART Connections
  - Arduino



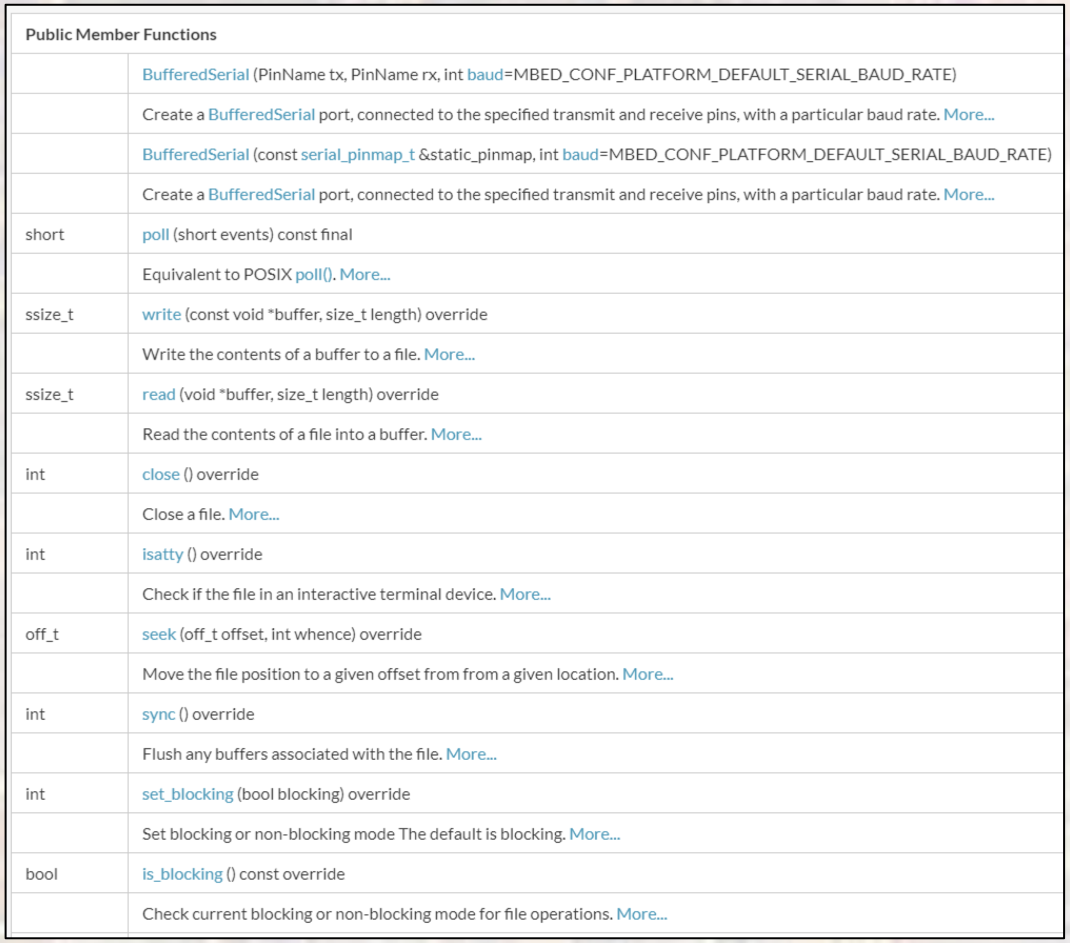
UART Programming

- UART Connections
  - Morpho



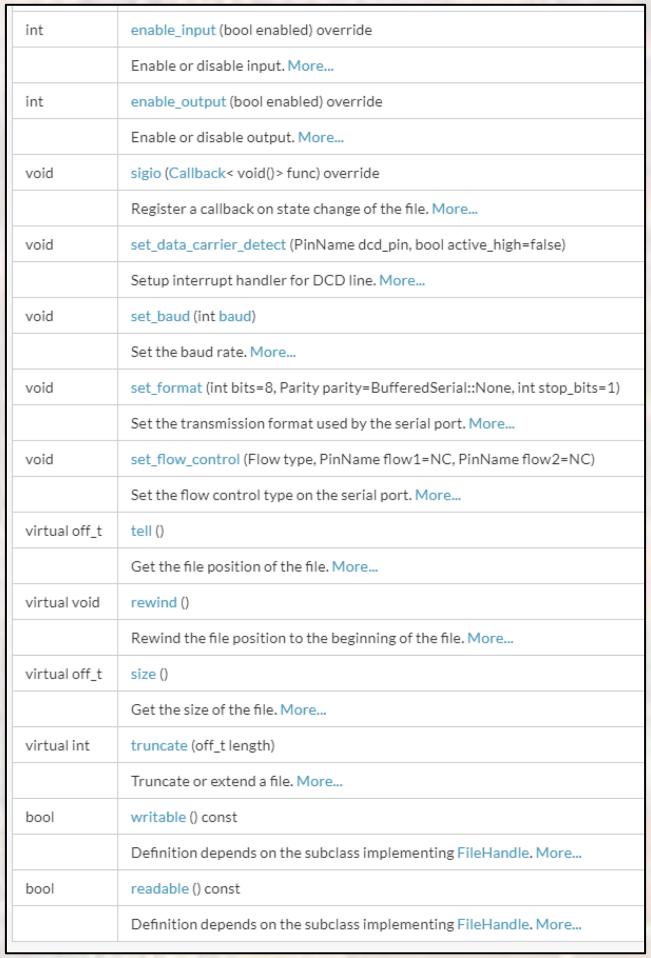
UART Programming

- BufferedSerial Class



Public Member Functions

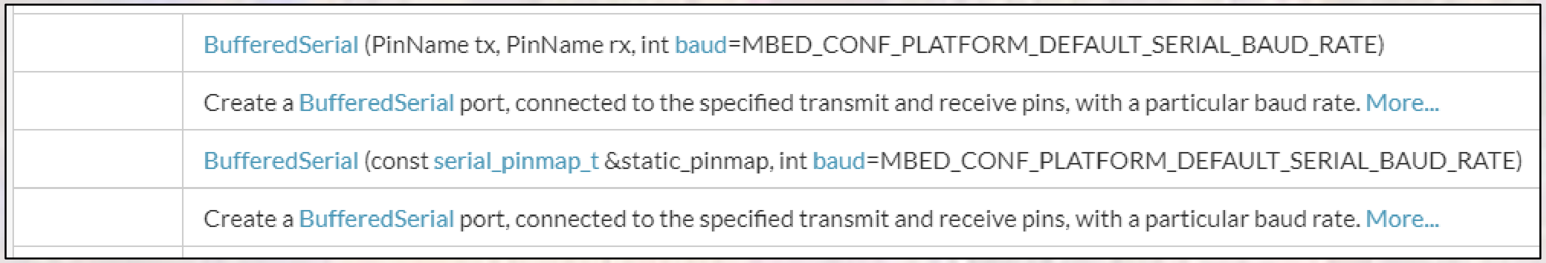
	BufferedSerial (PinName tx, PinName rx, int baud=MBED_CONF_PLATFORM_DEFAULT_SERIAL_BAUD_RATE)
	Create a BufferedSerial port, connected to the specified transmit and receive pins, with a particular baud rate. More...
	BufferedSerial (const serial_pinmap_t &static_pinmap, int baud=MBED_CONF_PLATFORM_DEFAULT_SERIAL_BAUD_RATE)
	Create a BufferedSerial port, connected to the specified transmit and receive pins, with a particular baud rate. More...
short	poll (short events) const final
	Equivalent to POSIX poll(). More...
ssize_t	write (const void *buffer, size_t length) override
	Write the contents of a buffer to a file. More...
ssize_t	read (void *buffer, size_t length) override
	Read the contents of a file into a buffer. More...
int	close () override
	Close a file. More...
int	isatty () override
	Check if the file in an interactive terminal device. More...
off_t	seek (off_t offset, int whence) override
	Move the file position to a given offset from from a given location. More...
int	sync () override
	Flush any buffers associated with the file. More...
int	set_blocking (bool blocking) override
	Set blocking or non-blocking mode The default is blocking. More...
bool	is_blocking () const override
	Check current blocking or non-blocking mode for file operations. More...



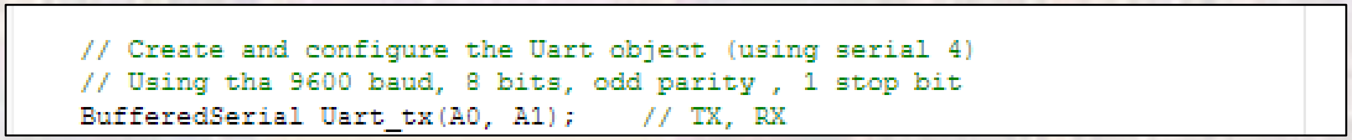
int	enable_input (bool enabled) override
	Enable or disable input. More...
int	enable_output (bool enabled) override
	Enable or disable output. More...
void	sigio (Callback< void()> func) override
	Register a callback on state change of the file. More...
void	set_data_carrier_detect (PinName dcd_pin, bool active_high=false)
	Setup interrupt handler for DCD line. More...
void	set_baud (int baud)
	Set the baud rate. More...
void	set_format (int bits=8, Parity parity=BufferedSerial::None, int stop_bits=1)
	Set the transmission format used by the serial port. More...
void	set_flow_control (Flow type, PinName flow1=NC, PinName flow2=NC)
	Set the flow control type on the serial port. More...
virtual off_t	tell ()
	Get the file position of the file. More...
virtual void	rewind ()
	Rewind the file position to the beginning of the file. More...
virtual off_t	size ()
	Get the size of the file. More...
virtual int	truncate (off_t length)
	Truncate or extend a file. More...
bool	writable () const
	Definition depends on the subclass implementing FileHandle. More...
bool	readable () const
	Definition depends on the subclass implementing FileHandle. More...

UART Programming

- Constructor



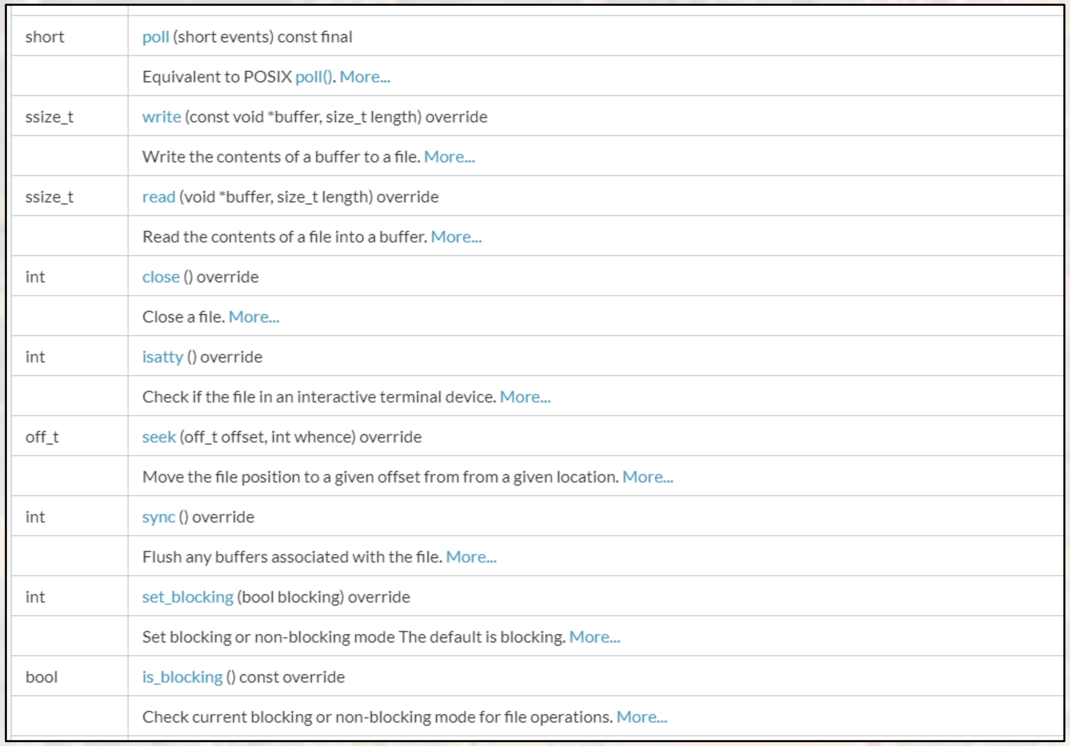
	BufferedSerial (PinName tx, PinName rx, int baud=MBED_CONF_PLATFORM_DEFAULT_SERIAL_BAUD_RATE)
	Create a BufferedSerial port, connected to the specified transmit and receive pins, with a particular baud rate. More...
	BufferedSerial (const serial_pinmap_t &static_pinmap, int baud=MBED_CONF_PLATFORM_DEFAULT_SERIAL_BAUD_RATE)
	Create a BufferedSerial port, connected to the specified transmit and receive pins, with a particular baud rate. More...



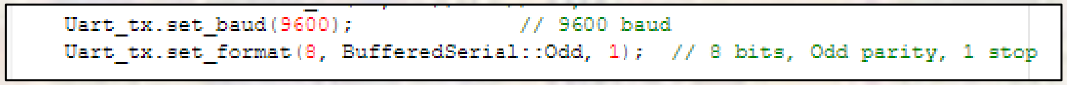
```
// Create and configure the Uart object (using serial 4)
// Using tha 9600 baud, 8 bits, odd parity , 1 stop bit
BufferedSerial Uart_tx(A0, A1);    // TX, RX
```

UART Programming

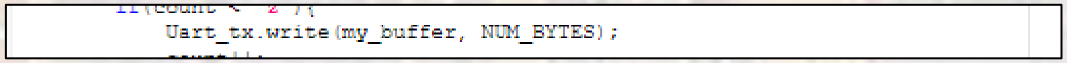
- Member Functions (Methods)



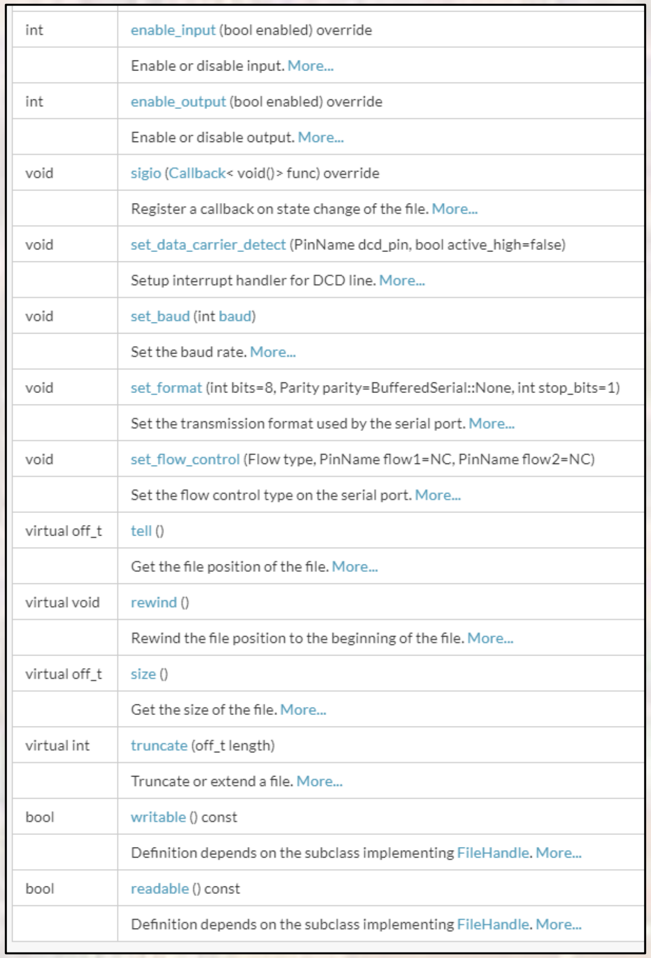
short	poll (short events) const final
	Equivalent to POSIX poll(). More...
ssize_t	write (const void *buffer, size_t length) override
	Write the contents of a buffer to a file. More...
ssize_t	read (void *buffer, size_t length) override
	Read the contents of a file into a buffer. More...
int	close () override
	Close a file. More...
int	isatty () override
	Check if the file in an interactive terminal device. More...
off_t	seek (off_t offset, int whence) override
	Move the file position to a given offset from from a given location. More...
int	sync () override
	Flush any buffers associated with the file. More...
int	set_blocking (bool blocking) override
	Set blocking or non-blocking mode The default is blocking. More...
bool	is_blocking () const override
	Check current blocking or non-blocking mode for file operations. More...



```
    Uart_tx.set_baud(9600);             // 9600 baud
    Uart_tx.set_format(8, BufferedSerial::Odd, 1);  // 8 bits, Odd parity, 1 stop
```



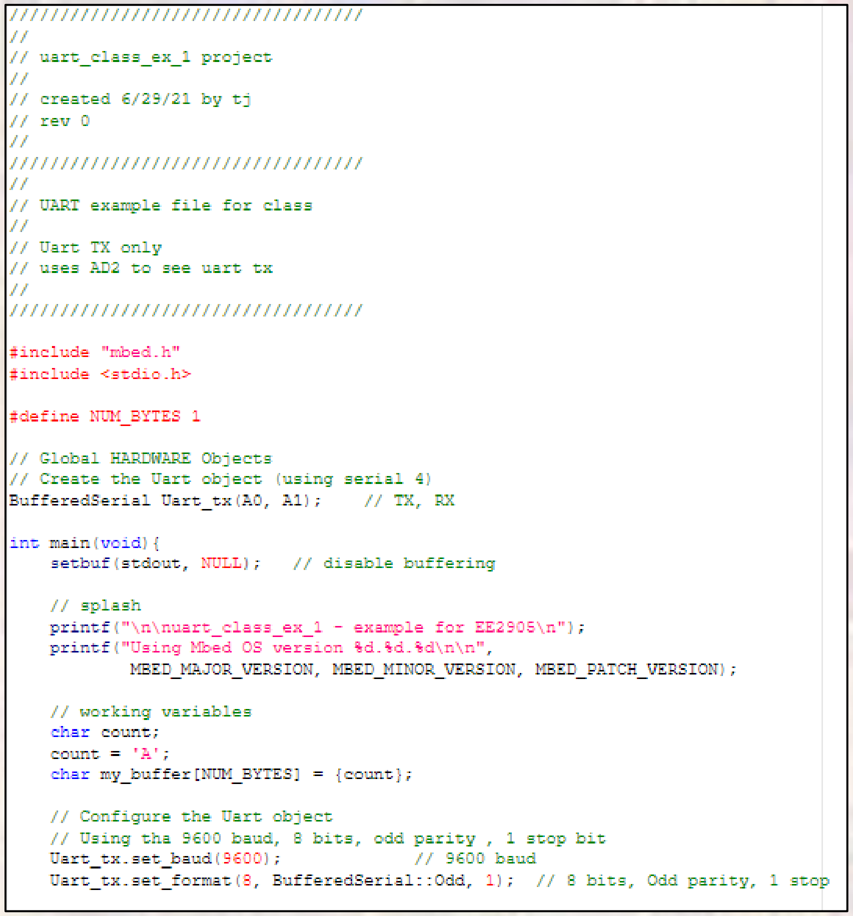
```
        Uart_tx.write(my_buffer, NUM_BYTES);
```



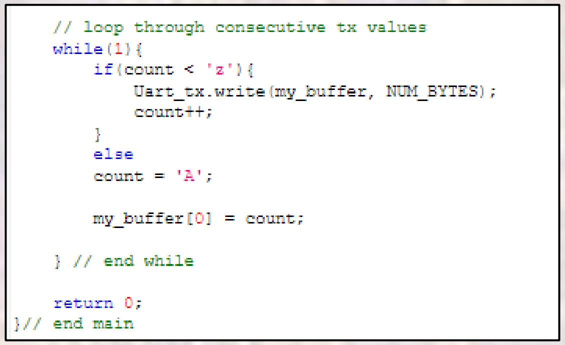
int	enable_input (bool enabled) override
	Enable or disable input. More...
int	enable_output (bool enabled) override
	Enable or disable output. More...
void	sigio (Callback< void()> func) override
	Register a callback on state change of the file. More...
void	set_data_carrier_detect (PinName dcd_pin, bool active_high=false)
	Setup interrupt handler for DCD line. More...
void	set_baud (int baud)
	Set the baud rate. More...
void	set_format (int bits=8, Parity parity=BufferedSerial::None, int stop_bits=1)
	Set the transmission format used by the serial port. More...
void	set_flow_control (Flow type, PinName flow1=NC, PinName flow2=NC)
	Set the flow control type on the serial port. More...
virtual off_t	tell ()
	Get the file position of the file. More...
virtual void	rewind ()
	Rewind the file position to the beginning of the file. More...
virtual off_t	size ()
	Get the size of the file. More...
virtual int	truncate (off_t length)
	Truncate or extend a file. More...
bool	writable () const
	Definition depends on the subclass implementing FileHandle. More...
bool	readable () const
	Definition depends on the subclass implementing FileHandle. More...

UART Programming

- Simple example 1
  - Tx



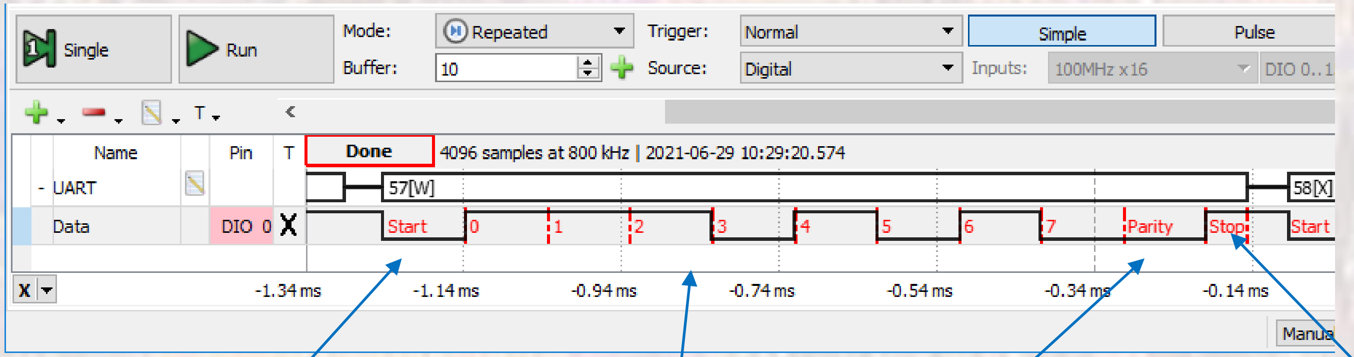
```
////////////////////////////////////
//
// uart_class_ex_1 project
//
// created 6/29/21 by tj
// rev 0
//
////////////////////////////////////
//
// UART example file for class
//
// Uart TX only
// uses AD2 to see uart tx
//
////////////////////////////////////

#include "mbed.h"
#include <stdio.h>

#define NUM_BYTES 1

// Global HARDWARE Objects
// Create the Uart object (using serial 4)
BufferedSerial Uart_tx(A0, A1);    // TX, RX

int main(void){
    setbuf(stdout, NULL);    // disable buffering

    // splash
    printf("\n\nuart_class_ex_1 - example for EE2905\n");
    printf("Using Mbed OS version %d.%d.%d\n\n",
            MBED_MAJOR_VERSION, MBED_MINOR_VERSION, MBED_PATCH_VERSION);

    // working variables
    char count;
    count = 'A';
    char my_buffer[NUM_BYTES] = {count};

    // Configure the Uart object
    // Using tha 9600 baud, 8 bits, odd parity , 1 stop bit
    Uart_tx.set_baud(9600);              // 9600 baud
    Uart_tx.set_format(8, BufferedSerial::Odd, 1);  // 8 bits, Odd parity, 1 stop
```



```
    // loop through consecutive tx values
    while(1){
        if(count < 'z'){
            Uart_tx.write(my_buffer, NUM_BYTES);
            count++;
        }
        else
        count = 'A';

        my_buffer[0] = count;

    } // end while

    return 0;
}// end main
```

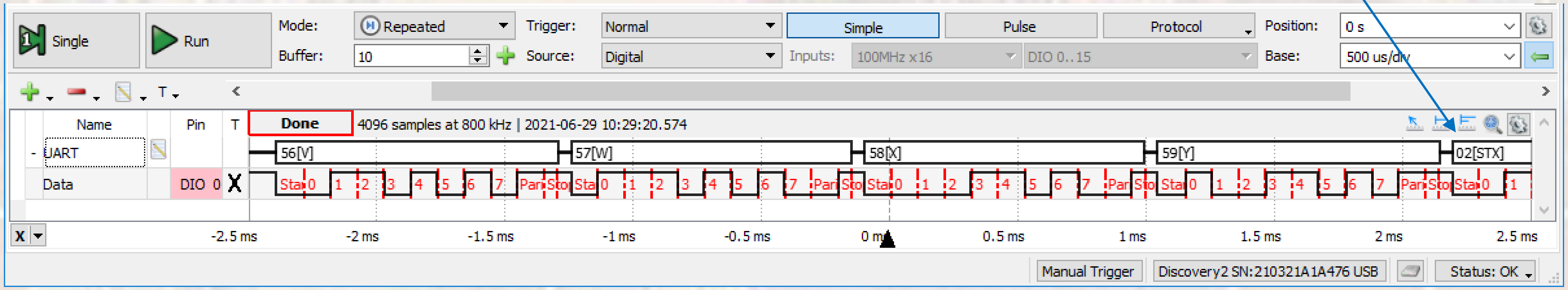
UART Programming

- Simple example 1
  - TX



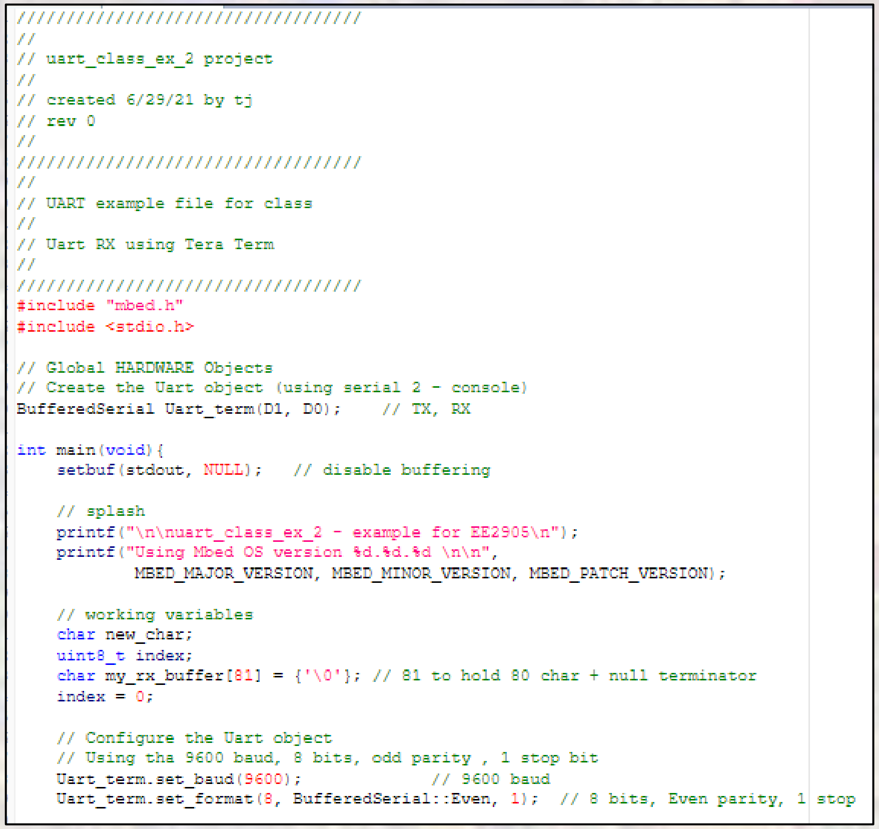
Captured – V – W – X - Y

Not an error: not enough info for waveforms to interpret the data

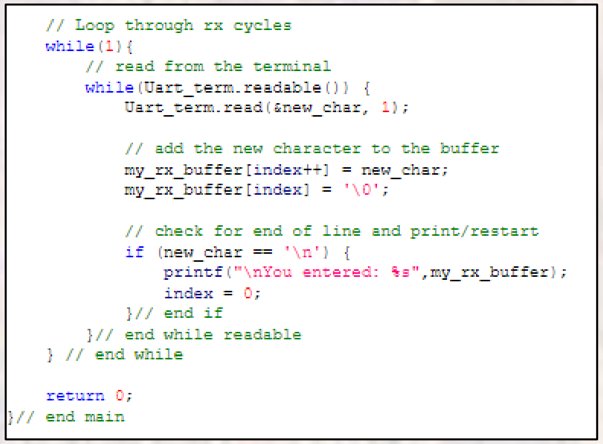


UART Programming

- Simple example 2
  - Rx



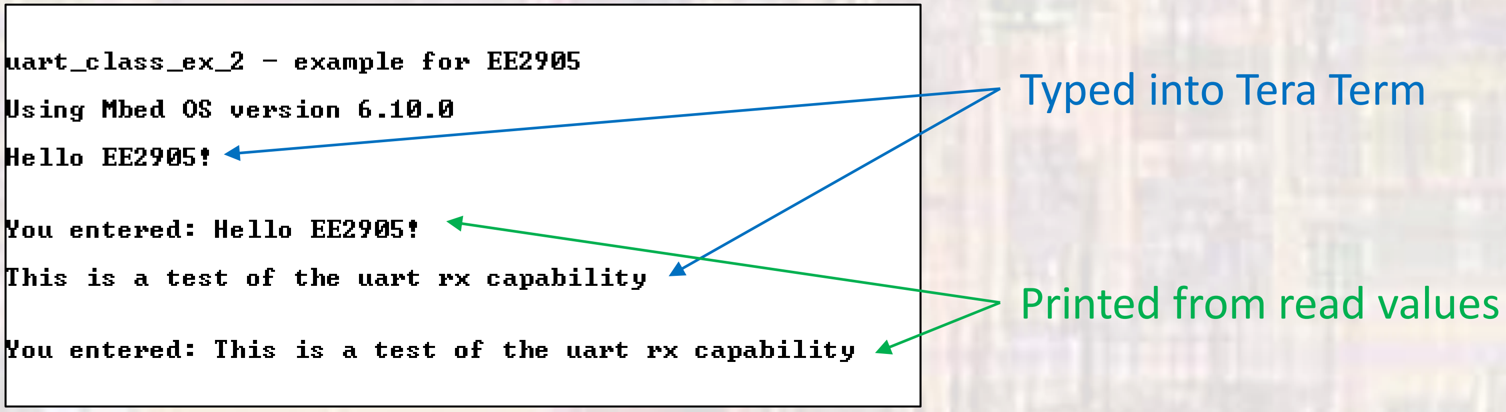
```
////////////////////////////////////
//
// uart_class_ex_2 project
//
// created 6/29/21 by tj
// rev 0
//
////////////////////////////////////
//
// UART example file for class
//
// Uart RX using Tera Term
//
////////////////////////////////////
#include "mbed.h"
#include <stdio.h>

// Global HARDWARE Objects
// Create the Uart object (using serial 2 - console)
BufferedSerial Uart_term(D1, D0);    // TX, RX

int main(void){
    setbuf(stdout, NULL);   // disable buffering

    // splash
    printf("\n\nuart_class_ex_2 - example for EE2905\n");
    printf("Using Mbed OS version %d.%d.%d \n\n",
            MBED_MAJOR_VERSION, MBED_MINOR_VERSION, MBED_PATCH_VERSION);

    // working variables
    char new_char;
    uint8_t index;
    char my_rx_buffer[81] = {'\0'}; // 81 to hold 80 char + null terminator
    index = 0;

    // Configure the Uart object
    // Using tha 9600 baud, 8 bits, odd parity , 1 stop bit
    Uart_term.set_baud(9600);                 // 9600 baud
    Uart_term.set_format(8, BufferedSerial::Even, 1);  // 8 bits, Even parity, 1 stop
```



```
    // Loop through rx cycles
    while(1){
        // read from the terminal
        while(Uart_term.readable()) {
            Uart_term.read(&new_char, 1);

            // add the new character to the buffer
            my_rx_buffer[index++] = new_char;
            my_rx_buffer[index] = '\0';

            // check for end of line and print/restart
            if (new_char == '\n') {
                printf("\nYou entered: %s",my_rx_buffer);
                index = 0;
            }// end if
        }// end while readable
    } // end while

    return 0;
}// end main
```

UART Programming

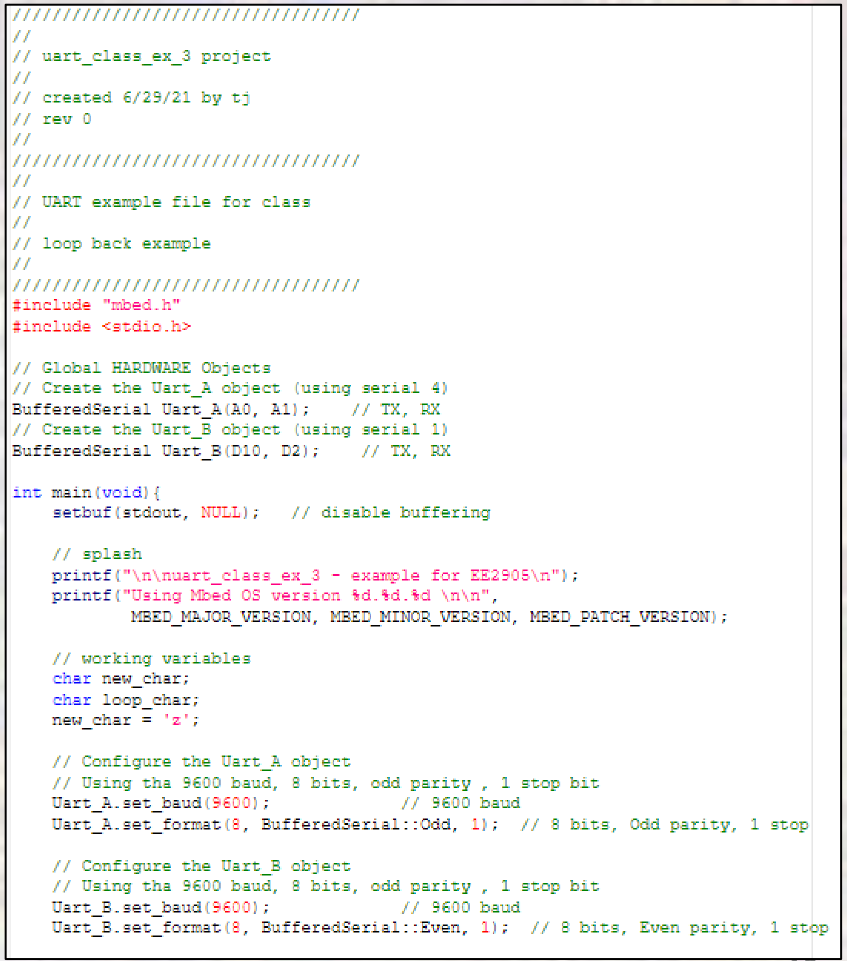
- Simple example 2
  - Rx



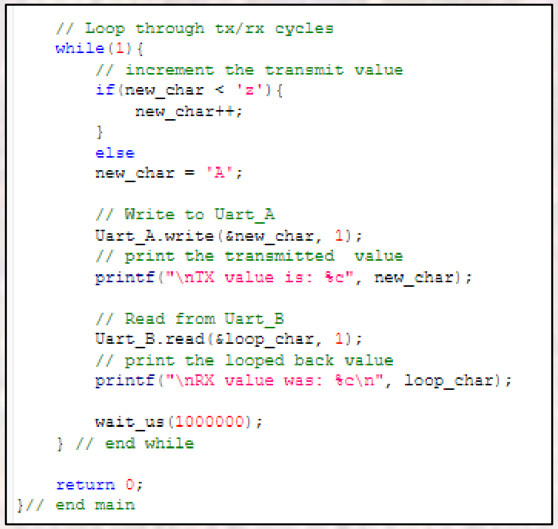
```
uart_class_ex_2 - example for EE2905
Using Mbed OS version 6.10.0
Hello EE2905!

You entered: Hello EE2905!
This is a test of the uart rx capability

You entered: This is a test of the uart rx capability
```

UART Programming

- Simple example 3
  - Loopback – connecting 2 UARTS on the board



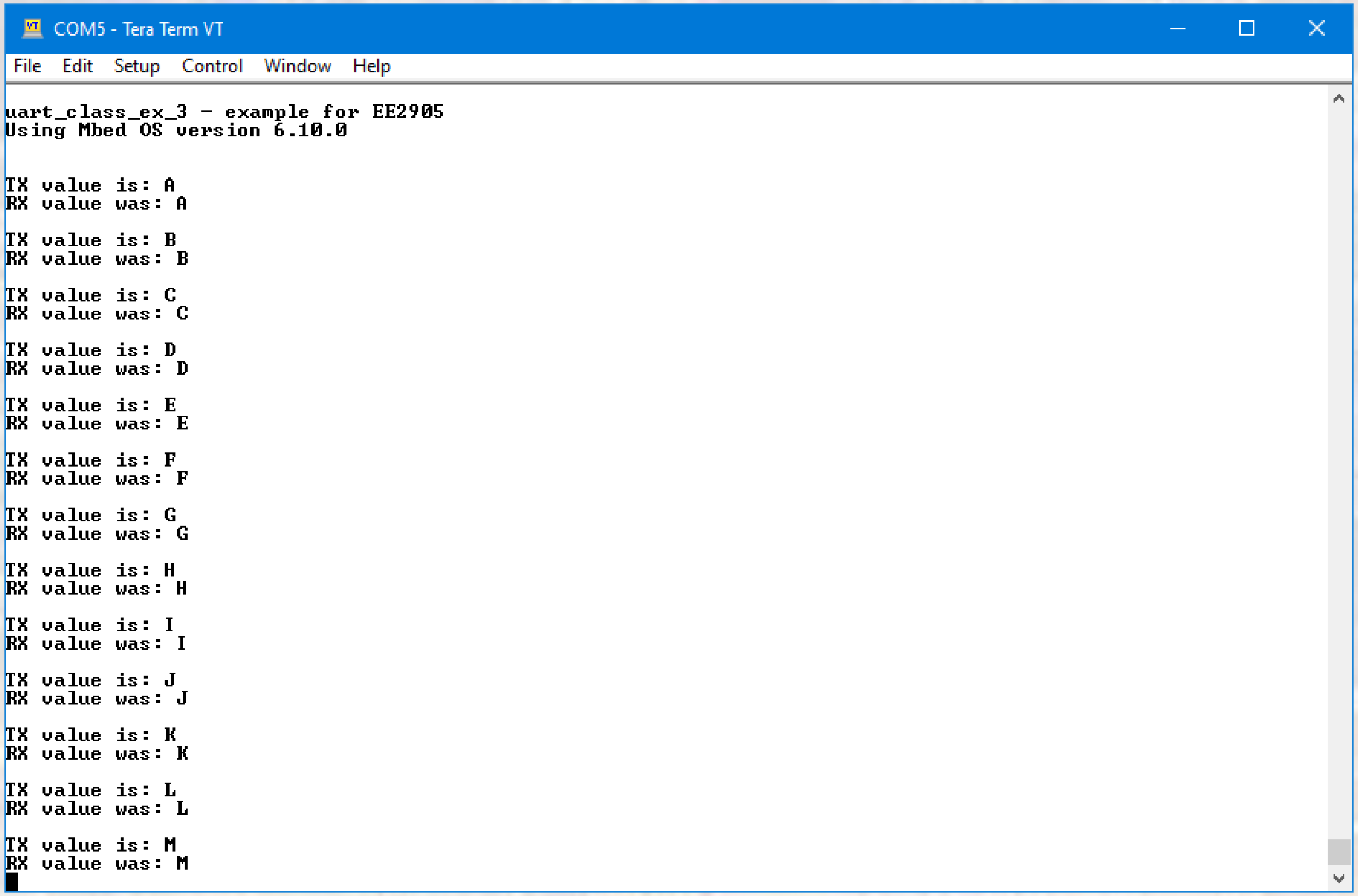
```
////////////////////////////////////
//
// uart_class_ex_3 project
//
// created 6/29/21 by tj
// rev 0
//
////////////////////////////////////
//
// UART example file for class
//
// loop back example
//
////////////////////////////////////
#include "mbed.h"
#include <stdio.h>

// Global HARDWARE Objects
// Create the Uart_A object (using serial 4)
BufferedSerial Uart_A(A0, A1);    // TX, RX
// Create the Uart_B object (using serial 1)
BufferedSerial Uart_B(D10, D2);    // TX, RX

int main(void){
    setbuf(stdout, NULL);   // disable buffering

    // splash
    printf("\n\nuart_class_ex_3 - example for EE2905\n");
    printf("Using Mbed OS version %d.%d.%d \n\n",
            MBED_MAJOR_VERSION, MBED_MINOR_VERSION, MBED_PATCH_VERSION);

    // working variables
    char new_char;
    char loop_char;
    new_char = 'z';

    // Configure the Uart_A object
    // Using tha 9600 baud, 8 bits, odd parity , 1 stop bit
    Uart_A.set_baud(9600);                // 9600 baud
    Uart_A.set_format(8, BufferedSerial::Odd, 1);  // 8 bits, Odd parity, 1 stop

    // Configure the Uart_B object
    // Using tha 9600 baud, 8 bits, odd parity , 1 stop bit
    Uart_B.set_baud(9600);                // 9600 baud
    Uart_B.set_format(8, BufferedSerial::Even, 1);  // 8 bits, Even parity, 1 stop
```



```
    // Loop through tx/rx cycles
    while(1){
        // increment the transmit value
        if(new_char < 'z'){
            new_char++;
        }
        else
        new_char = 'A';

        // Write to Uart_A
        Uart_A.write(&new_char, 1);
        // print the transmitted  value
        printf("\nTX value is: %c", new_char);

        // Read from Uart_B
        Uart_B.read(&loop_char, 1);
        // print the looped back value
        printf("\nRX value was: %c\n", loop_char);

        wait_us(1000000);
    } // end while

    return 0;
}// end main
```

UART Programming

- Simple example 3
  - Loopback – connecting 2 UARTS on the board



```
uart_class_ex_3 - example for EE2905
Using Mbed OS version 6.10.0


TX value is: A
RX value was: A

TX value is: B
RX value was: B

TX value is: C
RX value was: C

TX value is: D
RX value was: D

TX value is: E
RX value was: E

TX value is: F
RX value was: F

TX value is: G
RX value was: G

TX value is: H
RX value was: H

TX value is: I
RX value was: I

TX value is: J
RX value was: J

TX value is: K
RX value was: K

TX value is: L
RX value was: L

TX value is: M
RX value was: M
```

UART Programming

- Limitations summary

  - Baud rate * total bits transmitted determines transfer time