(1) 1.1 - access to board - GPIO writepin PA5 - turn on the onboard LED

(2) 1.2 - turn on and off the onboard LED at 1 second interval

(3) 1.3 - use togglepin function instead

(4) 1.4 lab - replicate above using access to mcu

(5) 1.5 lab - access to mcu - use any other pin apart from PA5. use writepin and then togglepin to see on and off of a pin using multimeter

(6) 2.1 hal gpio readpin - apply 3.3v to one of the pins. if 3.3 v applied to a pin, glow on board led, if 0 gnd applied, turn off the PA5 onboard LED

(7) 2.2 hal gpio readpin - push button (when pressed, voltage is low) pressed then only toggle the onboard led

(8) 2.3 lab - output 3.3v (writepin) and glow onboard led till the push button is pressed. produce 0v and turn off the led as soon as the push button is released

(9) 2.4 lab – repeat above with eternal led with resistor.

(10) 3.1 Toggle onboard LED everytime when blue push button is pressed. Use GPIO interrupt. Use Access to MCU selector.

(11) 3.2 lab - Write a program such that: if a blue push button is pressed once, the onboard LED starts toggling at 1 second. If a button is pressed again, the LED stops toggling. And this repeats. Note that Hal_Delay() function doesn't work in the interrupt (callback) routine. Hence you would have to use a flag in the interrupt routine and do corresponding actions (including the 1 sec delay) in the while(1) loop.

(12) 4.1 Transmit a string from STM to your computer's serial terminal. Use onboard UART2 connections. Use access to board selector.

(13) 4.2 Check the CubeMx UART configurations above and repeat the same with using access to MCU selector. Do not use interrupt.

(14) 4.3 Introduce UART reception in normal mode. Understand the problems that you are facing without the use of interrupt. Which problems?

(15) 4.4 Use UART reception with interrupt.

(16) 4.5 Use UART transmit and receive together and observe the bidirectional communication.

(17) 4.6 Repeat all of the above UART exercises in the lab.

(18) 4.7 Lab - Connect two STM boards and establish bidirectional communication using UART. One group transmits the data while the other group receives and vice a versa. Use external jumper wires and a different UART which is not configured onboard.

(19) 5.1 ADC in normal mode. Use PA0 (A0) pin.

(20) 5.2 ADC in interrupt mode - single measurement

(21) 5.3 ADC in interrupt mode - repeat measurement using CubeMx setting

(22) 5.4 ADC in DMA mode - single measurement

(23) 5.5 ADC in DMA mode - repeat measurement – restart ADC in callback function

(24) 5.6 lab - repeat all of the above

(25) 5.7 Learn how to use multichannel ADC

(26) 6.1 Use Timer 16 in interrupt mode to toggle onboard LED at 1 second.

(27) 6.2 lab – repeat above exercise with 2 second LED toggling interval

(28) 6.3 lab – use any other timer apart from timer 16 and repeat above.

(29) 6.4 lab – use 2 timers simultaneously to toggle on board LED at 1 second and to toggle any port pin voltage at 2 seconds interval.

(30) 6.5 lab – write a code to implement this - press on board push button -> timer starts and generates an interrupt after 1 second -> LED becomes on -> press push button again to turn it off