LEBANESE INTERNATIONAL UNIVERSITY

الجامعة اللبنانية الدولية

Lebanese International University

School of Engineering

Department of Computer and Communication

CENG400L—MicroController Lab

Section ID: F

Report ID: 3

Project Title: USART

Submitted by:Zainab Taleb

Lab Instructor:Ismail Kassem

Date:7/12/2023

School of Engineering

Department of Computer and Communication Engineering

CENG400L – Micro Lab

Report No. 3

Fall (23-24)

Thursday December 7 2023, 12:00

Student Name	Zainab Taleb
ID	92030141

	Grade	Weight
Objectives		10
Title		5
Cover		5
Equipment and Roles		10
Simulation Part		30
Hardware Part		30
Conclusion		10
Total		100

Instructions
Report submission is individually
Each Part includes: title (2 pts), description (10 pts), and procedures (10pts), results (8 pts).
Conclusion should contains five points, four related to each experiments+ Application point (4 pts/ each)
Please Don't delete or edit the Mark Scheme

I. Objectives of the Experiment (10points)

1. Introduce Universal Synchronous Asynchronous Receiver Transmitter Serial (USART) port
2. Send and receive Data through AVR serially.
3. What does USART stands for?
4. What pins are responsible for transmitting and receiving data through ATMEGA328P?
5. Create serial communication between 2 ATmega328p.

II. Equipment used with role of each (10 Points)

1-ATmega328p:is a microcontroller chip from the AVR family, developed by ATMEL. It is widely used in various electronic projects and applications.

2- Potentiometer: referred to as a "pot" is a three-terminal variable resistor with an adjustable or rotary knob. It is commonly used in electronic circuits to control the voltage or signal level.

3-Arduino Uno: is a popular and widely used open-source microcontroller development board. It includes: digital and analog pins, microcontroller, usb interface, clock speed, input voltage operating voltage, IDE ,open-source platform, extensible.

4-wires: to connect the components together.

5-resistor: passive electronic components ,it is limiting or controlling the flow of electrical current in a circuit .

6-LED:leds for checking the working of the circuit.

7-PUSH button: to pressed when it needed.pressed➔power on, not presses➔off.

III. Simulation Part: USART (30 pts for the whole experiment)

A. Description

USART: stands for Universal Synchronous Asynchronous Receiver Transmitter. It is sometimes called the Serial Communications Interface or SCI. The USART is most commonly used in the asynchronous mode to communicate to a PC serial port. USART can be configured using four main registers in atmega328P microcontroller.

USART Control and Status Register 0 A (UCSR0A)

Bit	7	6	5	4	3	2	1	0
	RXC0	TXC0	UDRE0	FE0	DOR0	UPE0	U2X0	MPCM0
Access	R	R/W	R	R	R	R	R/W	R/W
Reset	0	0	1	0	0	0	0	0

USART Control and Status Register 0 B (UCSR0B)

Bit	7	6	5	4	3	2	1	0
	RXCIE0	TXCIE0	UDRIE0	RXEN0	TXEN0	UCSZ02	RXB80	TXB80
Access	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W
Reset	0	0	0	0	0	0	0	0

USART Control and Status Register 0 C (UCSR0C)

Bit	7	6	5	4	3	2	1	0
	UMSEL01	UMSEL00	UPM01	UPM00	USBS0	UCSZ01 / UDORD0	UCSZ00 / UCPHA0	UCPOL0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	1	1	0

B.Procedure:

1. In proteus ,design the circuit
2. Then write the code of TX and RX in 2 different files in Atmel Studio.
3. Check the COM of each chip
4. Put each code on the specific Atmega328p in proteus .
5. Run the program.

File Edit Format View Help

```
//TX
include "m328pdef.inc"
.org 0x00
LDI R17,0X67// R17 = LOW (UBRRN calculated in step 3)
STS UBRR0L,R17// UBRR0L = R17
LDI R17,0X00// R17 = HIGH (UBRRN rate calculated in step 3)
STS UBRR0H,R17// UBRR0H = R17

LDI R17,0b00001000/*
R17 = 8-bit value which is the content of UCSR0B register (Page 43 manual book)
MSB --> LSB:
bit 7: (Disable interrupt on the RXC0 Flag)
bit 6: (Disable interrupt on the TXC0 Flag)
bit 5: (Disable interrupt on the UDRE0 Flag)
bit 4: (Disable the USART Receiver)
bit 3: (enables the USART Transmitter)
bit 2: (Ch size = 8-bits)
bit 1: 0
bit 0: 0
*/
STS UCSR0B,R17// UCSR0B = R17
LDI R17,0b00001011/*
R17 = 8-bit value which is the content of UCSR0C register (Page 43 manual book)
MSB --> LSB:
bit 7-6: (Async mode)
bit 5-4: (Parity mode disable)
bit 3 : (Stop bit = 1)
bit 2-1: (Character size = 8-bits)
bit 0 : 0
*/
STS UCSR0C,R17// UCSR0C = R17
INITMEM:
LDI ZH,HIGH(MEM<<1)
LDI ZL,LOW(MEM<<1)
LOOP:
call delay
```

```
LOOP:
call delay
LPM R19,Z+// Load Program Memory into R19 register and Post-Increment Z
CPI R19,0X04// Compare r19 with 0x04
BREQ INITMEM// if r19 == 0x04 branch to INITMEM subroutine
Loop1:
LDS R17,UCSR0A// R17 = UCSR0A
SBRS R17,5//skip next instruction if the bit number 5 in r17 (UDRE0) is set
JMP LOOP1
STS UDR0,R19// UDR0 = R19
sendend:
LDS R17,UCSR0A// R17 = UCSR0A
SBRS R17,6//skip next instruction if the bit number 6 in r17 (TXC0) is set
JMP sendend
JMP LOOP
delay:
LDI R16,HIGH(20000)
STS TCNT1H,R16
LDI R16,LOW(20000)

STS TCNT1L,R16
LDI R16,0
STS TCCR1A,R16
LDI R16,4
STS TCCR1B,R16
again:
SBIS TIFR1,TOV1
JMP again
LDI R16,0
STS TCCR1B,R16
SBI TIFR1,TOV1
RET
.org 0x500
MEM:
.db 0x00,0x01,0x02,0x03,0x04/////////
```

RX:

```
include "m328pdef.inc"
.org 0x00
SBI DDRB,0// B0 OUTPUTvc
SBI DDRB,1// B1 OUTPUT
LDI R17,0X67// R17 = LOW (UBRRN calculated in step 3)
STS UBRR0L,R17// UBRR0L = R17
LDI R17,0X00// R17 = HIGH (UBRRN calculated in step 3)
STS UBRR0H,R17// UBRR0H = R17
LDI R17,0b00010000/*
R17 = 8-bit value which is the content of UCSR0B register (Page 43 manual book)
MSB --> LSB:
bit 7: (Disable interrupt on the RXC0 Flag)
bit 6: (Disable interrupt on the TXC0 Flag)
bit 5: (Disable interrupt on the UDRE0 Flag)
bit 4: (enable the USART Receiver)
bit 3: (Disable the USART Transmitter)
bit 2: (Ch size = 8-bits)
bit 1: 0
bit 0: 0
*/
STS UCSR0B,R17// UCSR0B = R17
LDI R17,0b00000110/*
R17 = 8-bit value which is the content of UCSR0C register (Page 43 manual book)
MSB --> LSB:
bit 7-6: (Async mode)
bit 5-4: (Parity mode disable)
bit 3 : (Stop bit = 1)
bit 2-1: (Character size = 8-bits)
bit 0 : 0
*/

STS UCSR0C,R17// UCSR0C = R17
RECEIVECOMPLETE:
LDS R17,UCSR0A// R17 = UCSR0A
SBRS R17,7// skip next instruction if bit 7 ( RXC0 ) in R17 is set
JMP RECEIVECOMPLETE
```

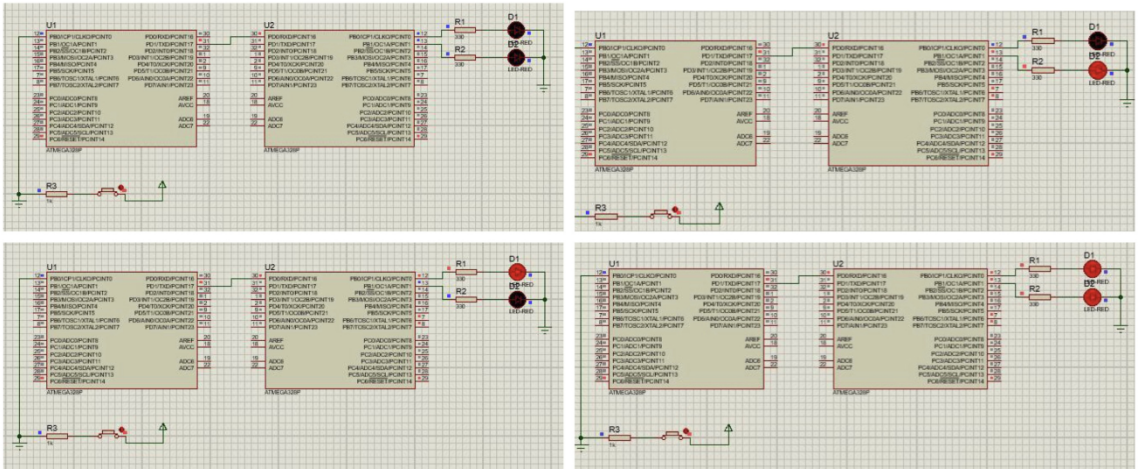
```
STS UCSR0C,R17// UCSR0C = R17
RECEIVECOMPLETE:
LDS R17,UCSR0A// R17 = UCSR0A
SBRS R17,7// skip next instruction if bit 7 ( RXC0 ) in R17 is set
JMP RECEIVECOMPLETE
LDS R19,UDR0// R19 = UDR0
OUT PORTB,R19// PORTB = R19
JMP RECEIVECOMPLETE

delay:
LDI R16,HIGH(3036)
STS TCNT1H,R16
LDI R16,LOW(3036)
STS TCNT1L,R16
LDI R16,0
STS TCCR1A,R16
LDI R16,4
STS TCCR1B,R16
again:
SBIS TIFR1,TOV1
JMP again
LDI R16,0
STS TCCR1B,R16
SBI TIFR1,TOV1
RET
```

Some registers used:

LDI:Load Immediate
STS:store string
SBI:set bit in I/O registerb
UDRR0H: for reading and writing data
UCSR0A:Register related to USART module
SBIS:Skip if Bit I/O register is set
DDR: data direction registers
SBRS: skip if bit in register is set

B. Results

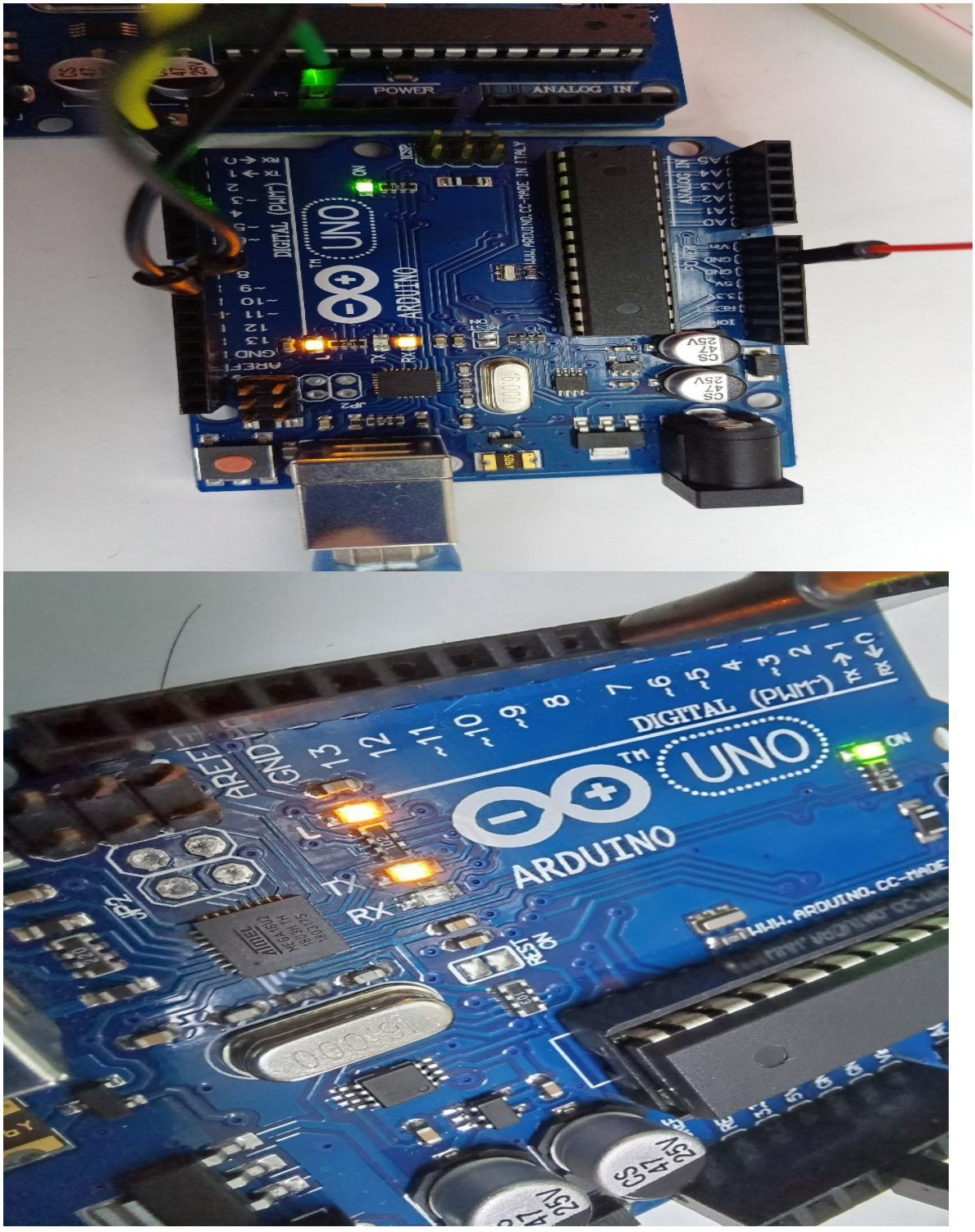
This is the result of the simulation part in proteus



IV. Hardware Part: USART hardware (30 pts for the whole experiment)

A. Description

In this experiment we are requested to create a serial communication between 2 atmega328p, where the TX keeps reading the status of B0 , if the push button is pressed , it sends the value of count to RX to display it on 2 leds connected to portb , Else, if the counter is = 3 , the counter will reset to 0.
When connecting the two chip together we obtain this result on hardware:

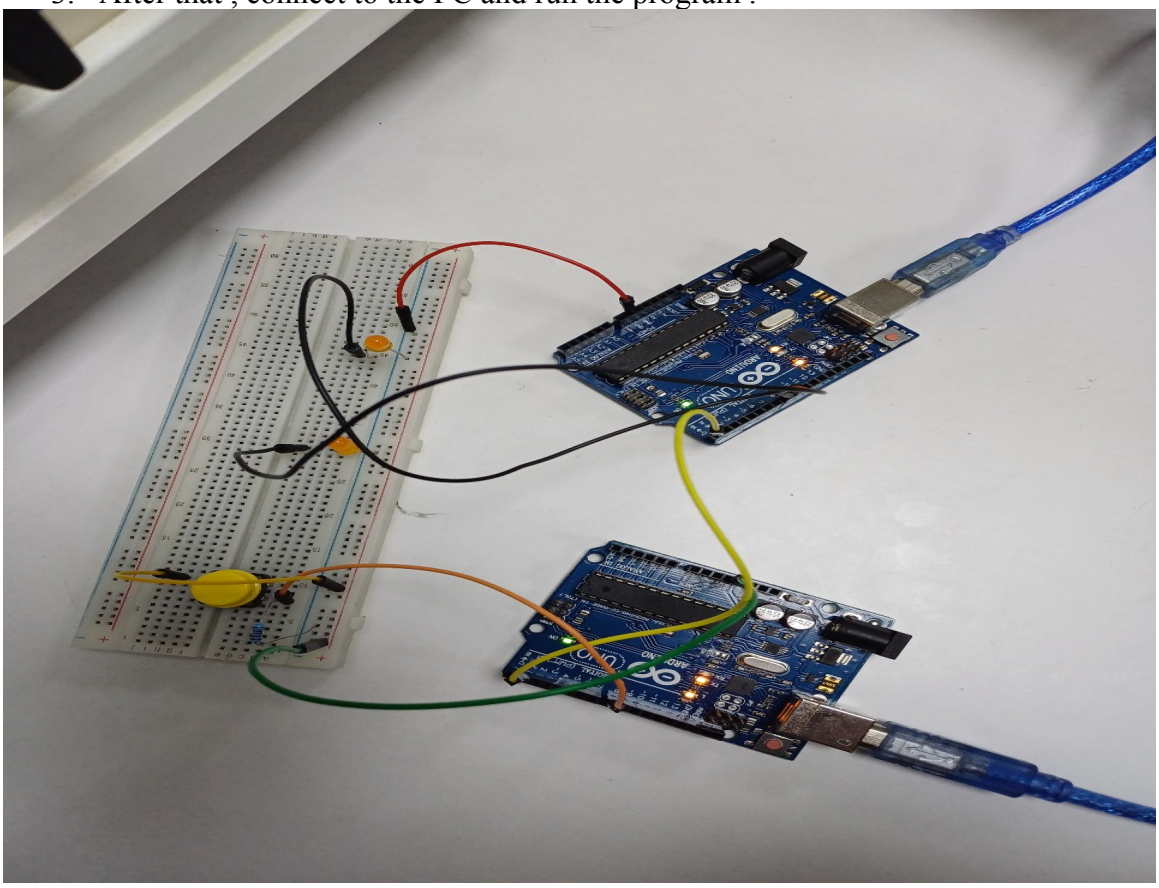


In the first RX is ON ,The second TX is ON.

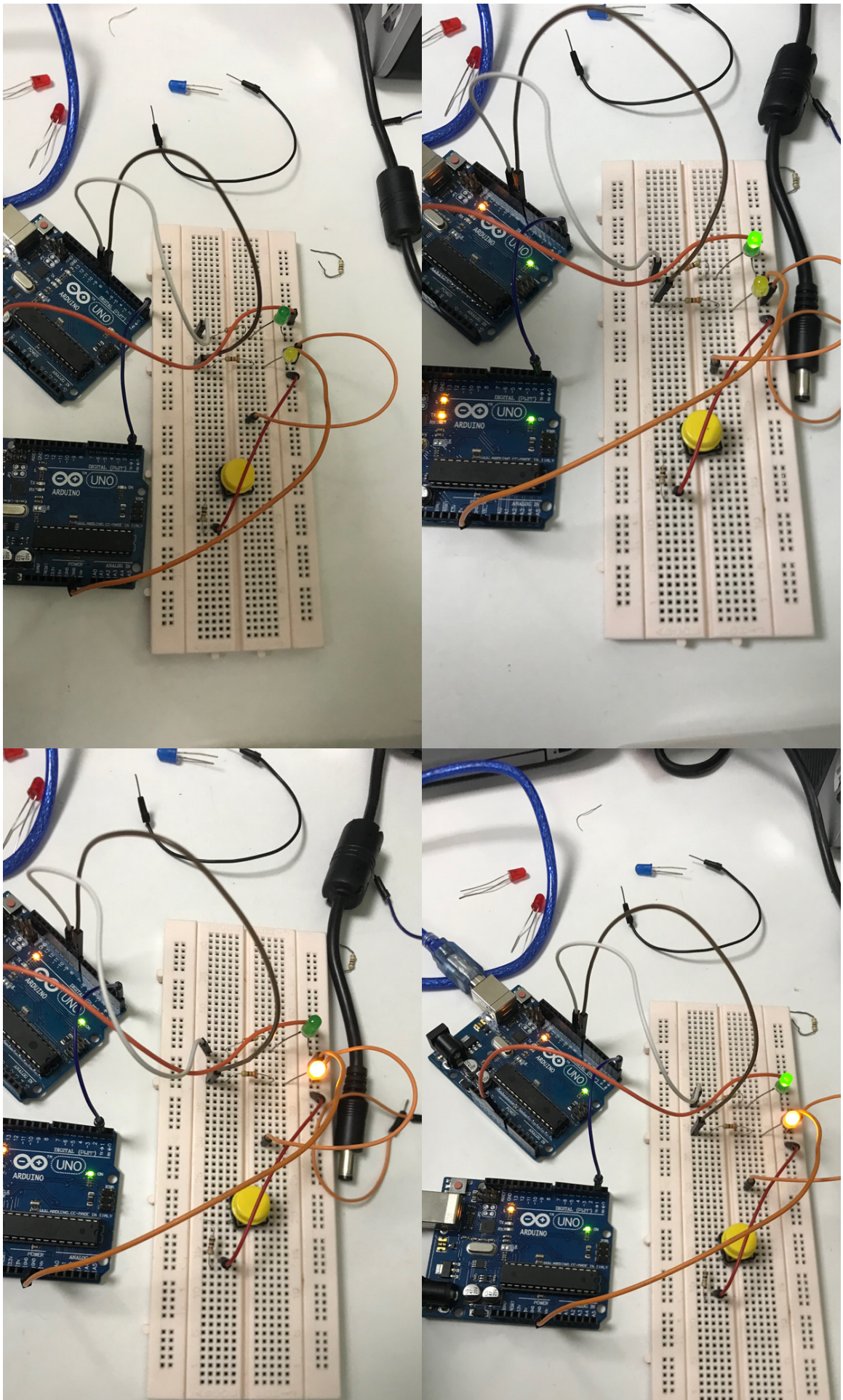
B. Procedure :

1. Put on the bread board the essential components: push button,leds ,resistors.
2. Connect them using wires.

3. After that , connect to the PC and run the program .



C. Results


ARDUINO
UNO

VII. Conclusion (10 Points)

1. USART is frequently used for communication between microcontrollers and peripherals such as sensors, displays, and memory devices. It provides a simple and efficient way for these devices to exchange data.
2. The Rx pin is the input for receiving data. It is connected to the Tx (Transmit) pin of the transmitting device. When data is sent from the transmitting device, it is received by the Rx pin of the receiving device. The receiving device's USART or UART module processes the incoming serial data, making it available for the microcontroller or other components to use.
3. The Tx pin is the output for transmitting data. It is connected to the Rx (Receive) pin of the receiving device. When data needs to be sent from the transmitting device to the receiving device, it is transmitted through the Tx pin. The USART or UART module in the transmitting device converts parallel data (from a microcontroller or other sources) into serial data and sends it through the Tx pin.
4. Using a push button in conjunction with USART (Universal Synchronous Asynchronous Receiver Transmitter) typically involves a microcontroller or embedded system where you want to trigger a USART-related action, such as sending data, when the push button is pressed.
5. TXC0 &RXC0 are used to check if the USART transmitting and receiving process are completed done.