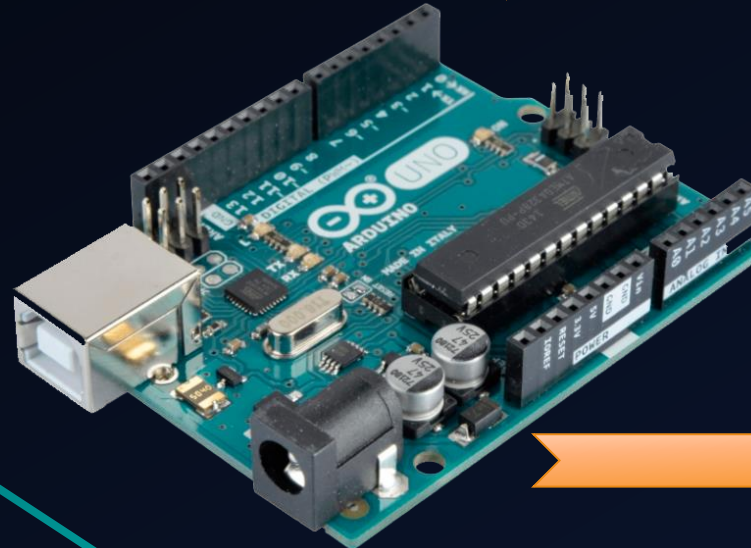
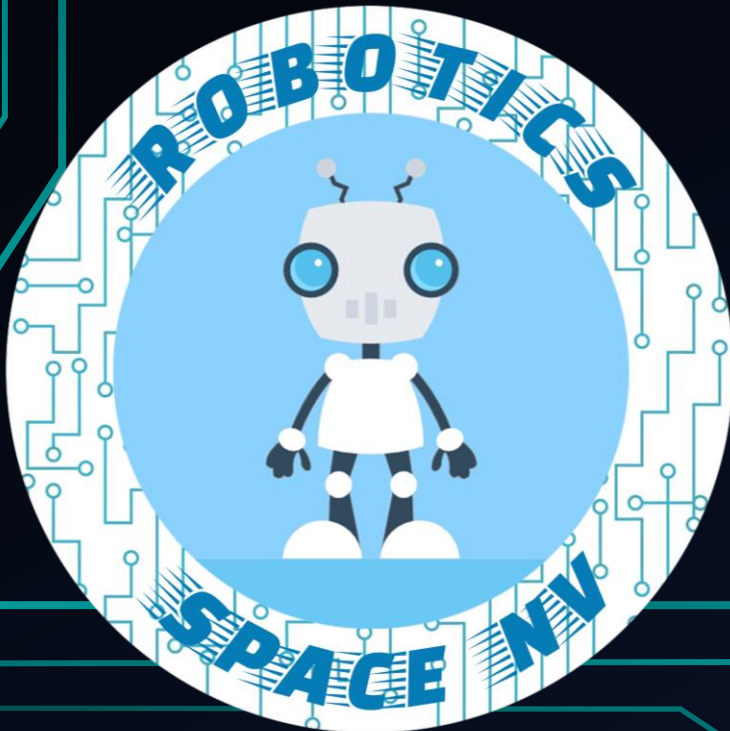


Clase 8

COMUNICACIÓN SERIAL Y GRÁFICAS

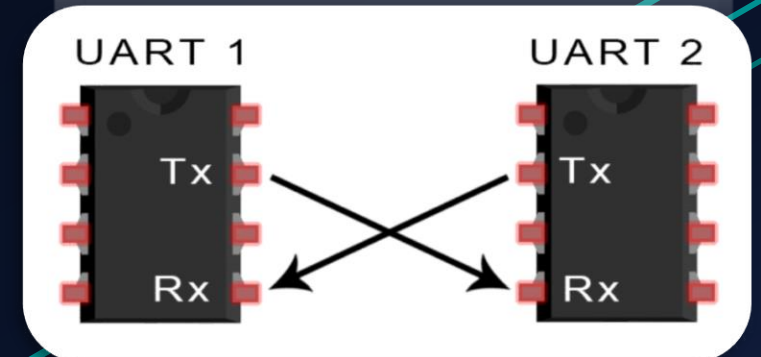
Suscríbete



```
sketch_feb10a Arduino 1.8.5
Archivo Editor Programa Herramientas Ayuda

sketch_feb10a

1 void setup() {
2   // put your setup code here, to run once:
3
4 }
5
6 void loop() {
7   // put your main code here, to run repeatedly:
8
9 }
```



TUTOR:NAGIB LUIS VALLEJOS M.

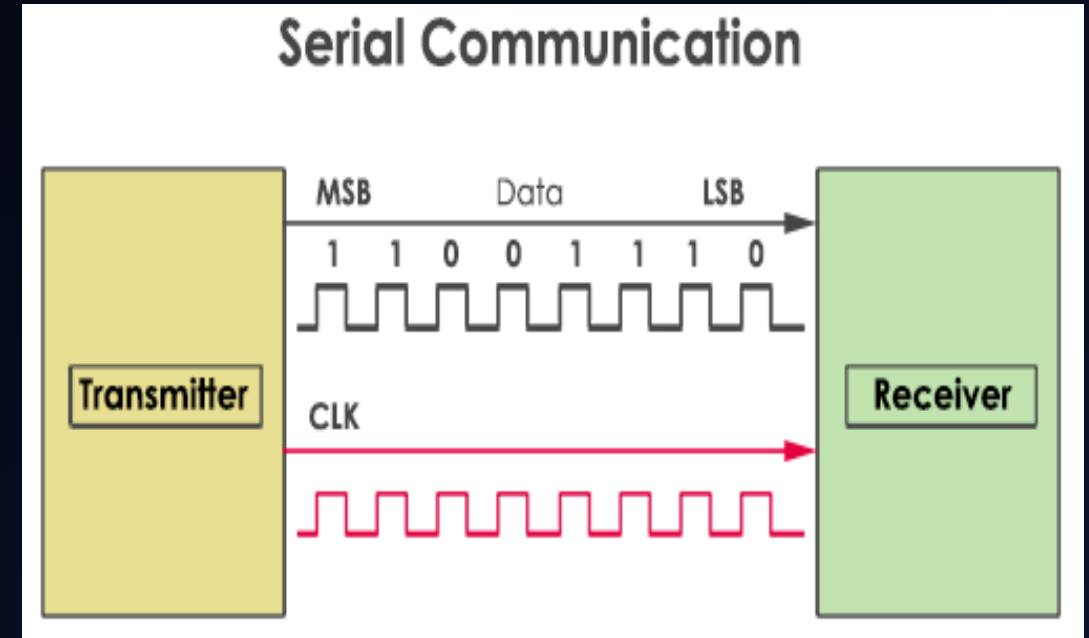


COMUNICACIÓN SERIAL

La comunicación serial, conocida como protocolo UART, es un canal de comunicación que nos permite conectar nuestras placas arduino con otros dispositivos, generando una comunicación bidireccional. Cuenta con 2 pines de conexión los cuales son:

TX y RX.

Todas las placas arduino cuenta por lo menos con 1 puerto serie para trabajar a través de este protocolo.



TUTOR:NAGIB LUIS VALLEJOS M.

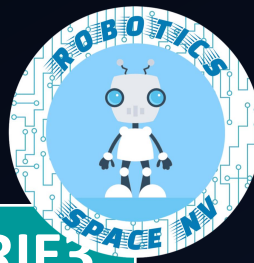
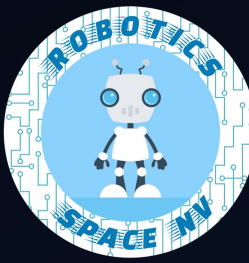


TABLA DE PLACAS POR CANTIDAD DE PS

PLACA	PINES SERIALES	PINES SERIE1	PINES SERIE2	PINES SERIE3
Uno, nano, mini	RX=0 – TX=1			
Mega	RX=0 – TX=1	RX=19 – TX=18	RX=17 – TX=16	RX=15 – TX=14
Leonardo, micro, Yun		RX=0 – TX=1		
Uno WiFi	Conectado por usb	RX=0 – TX=1	Conectado a niña	
Familia MKR		RX=13 – TX=14		
Zero	Conectado al port prog	RX=0 – TX=1		
Due	RX=0 – TX=1	RX=19 – TX=18	RX=17 – TX=16	RX=15 – TX=14
101		RX=0 – TX=1		

En arduino uno, nano, mega y mini los pines 0 y 1 se utilizan para comunicarse con la computadora

TUTOR:NAGIB LUIS VALLEJOS M.

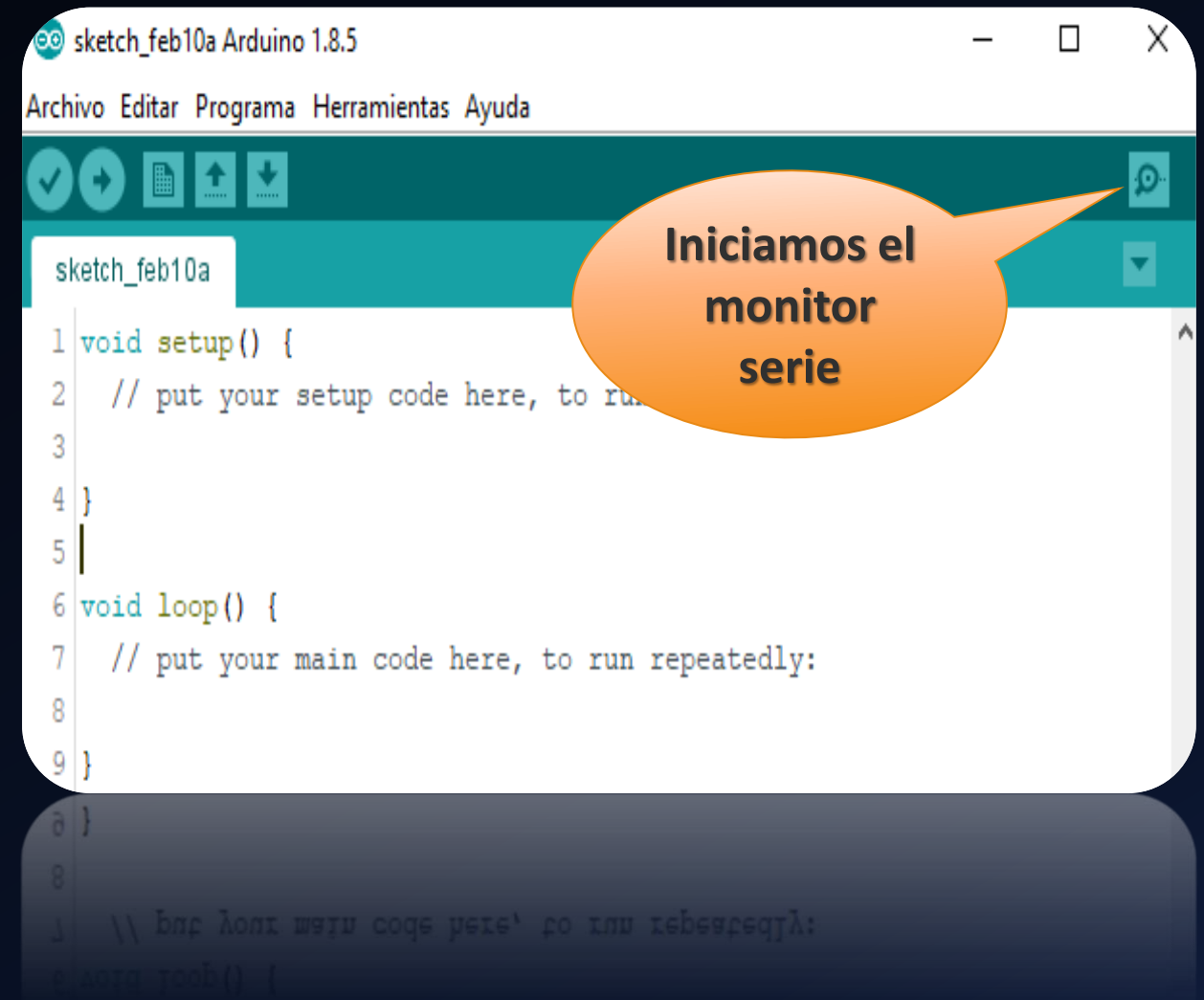


MONITOR SERIE

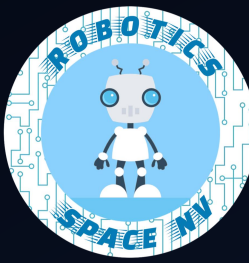
Es una interfaz visual con la que cuenta arduino, a través del monitor serie se puede imprimir valores de diferentes variables y señales, también podemos ingresar datos desde el teclado y visualizarlos

Para poder trabajar con el monitor serie se lo inicializa con la función:

Serial.begin(tasa de baudios);



TUTOR:NAGIB LUIS VALLEJOS M.



FUNCIONES ESENCIALES

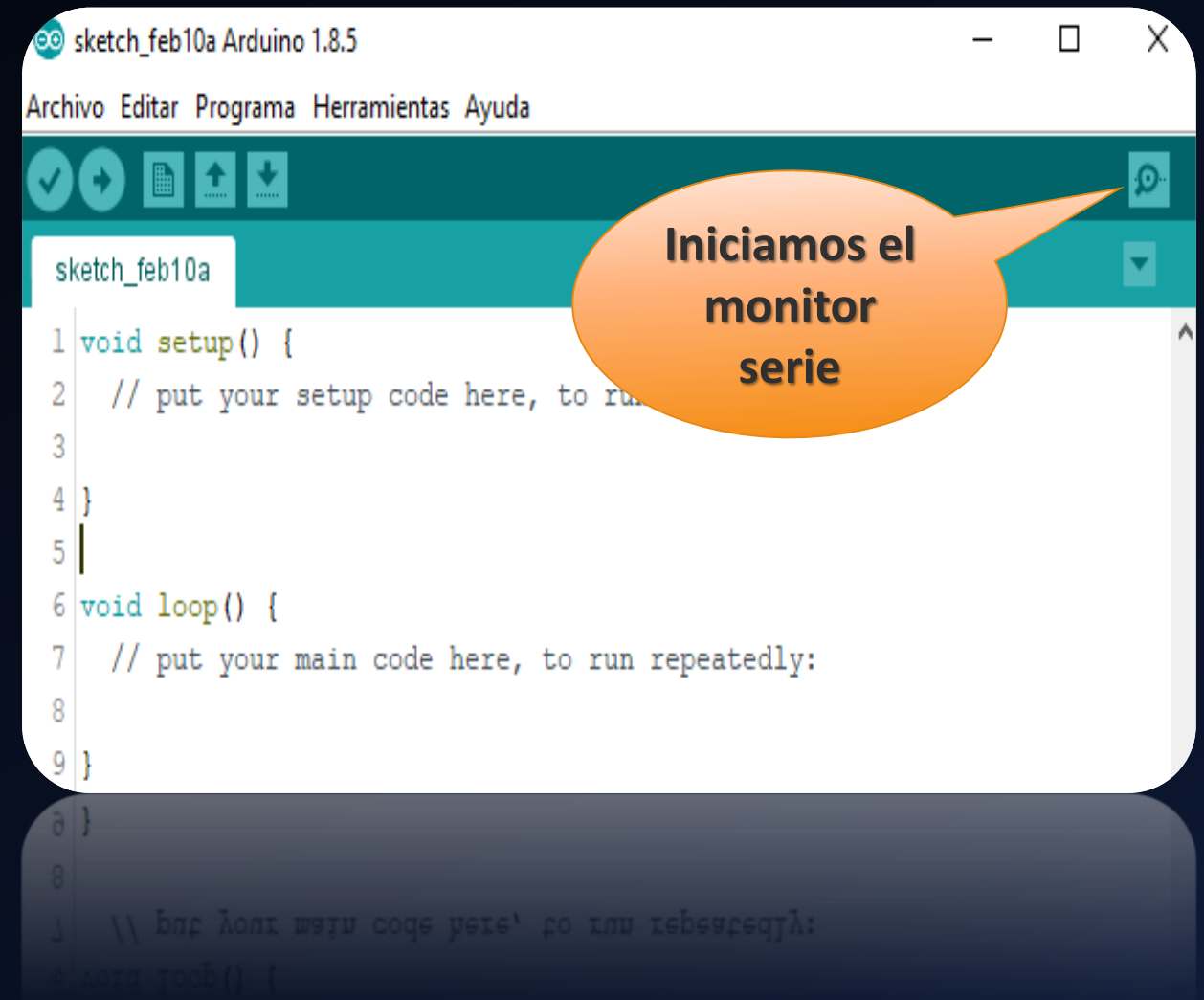
Serial.print(): Permite imprimir datos de manera continua.

Serial.println(): Permite imprimir datos acompañados de un salto de línea.

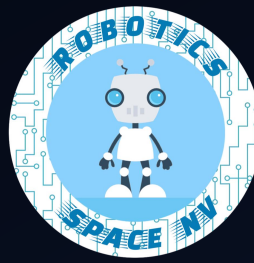
Serial.write(): Permite enviar datos a través del monitor serie a otros entornos.

Serial.read(): Permite leer datos que llegan a través del monitor serie.

Serial.end()



TUTOR:NAGIB LUIS VALLEJOS M.



EJEMPLO 1

Imprimir el mensaje “HolaMundo” empleando `Serial.print()` y `Serial.println()` a razón de un segundo.

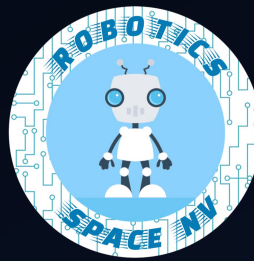
S8-E1

```
1 void setup() {  
2   Serial.begin(9600);  
3 }  
4 void loop() {  
5   Serial.print("Hola");  
6   delay(1000);  
7   Serial.println("Mundo");  
8   delay(1000);  
9 }
```

COM33 (Arduino Leonardo)

HolaMundo
HolaMundo
HolaMundo
HolaMundo
HolaMundo
HolaMundo
HolaMundo

TUTOR:NAGIB LUIS VALLEJOS M.



EJEMPLO 2

Imprimir su nombre completo empleando el siguiente formato, a razón de un segundo por impresión

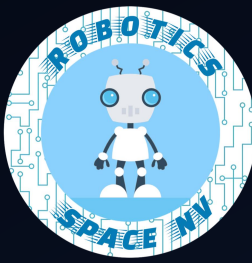
```
S8-E2
1 void setup() {
2   Serial.begin(9600);
3 }
4 void loop() {
5   Serial.println("Nagib");
6   delay(1000);
7   Serial.println("Luis");
8   delay(1000);
9   Serial.println("Vallejos");
10  delay(1000);
11  Serial.println("Mamani");
12  delay(1000);
13  Serial.println("");
14  delay(1000);
15 }
```

COM33 (Arduino Leonardo)

```
Nagib
Luis
Vallejos
Mamani

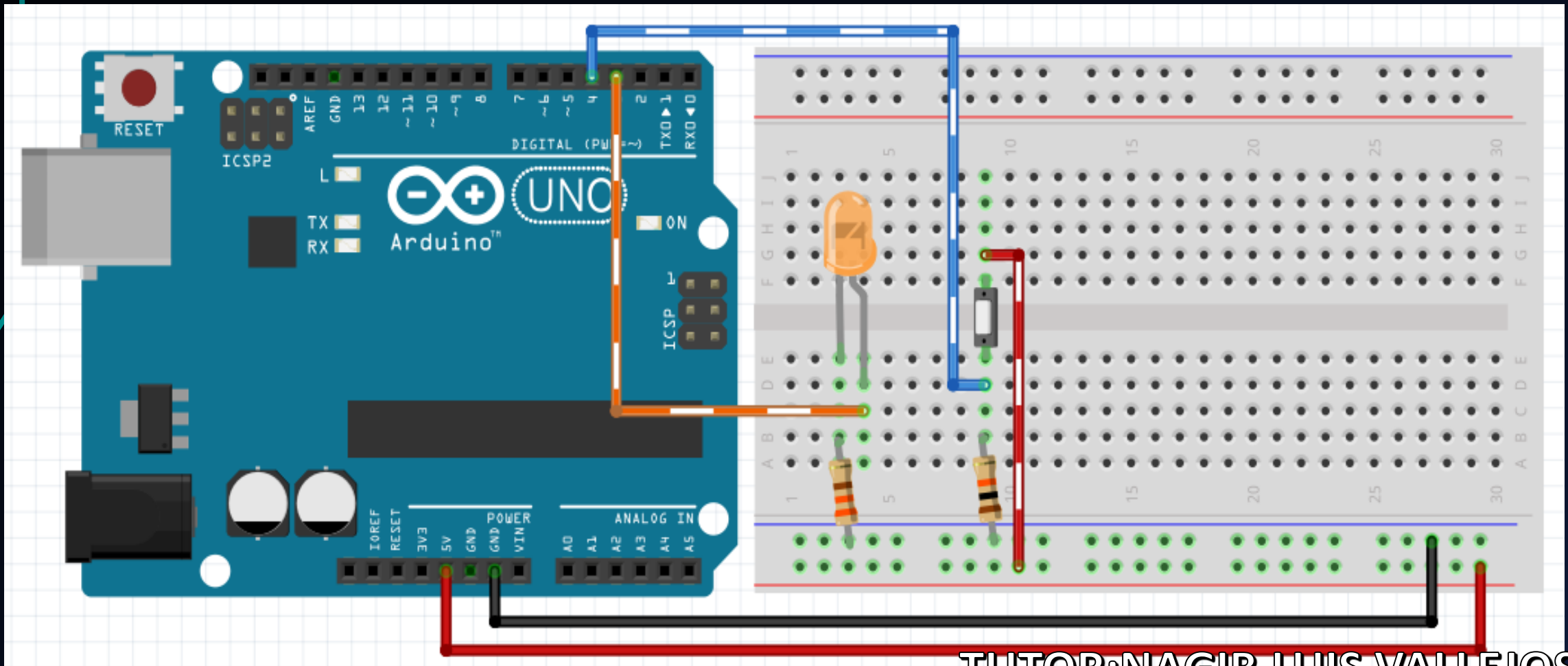
Nagib
Luis
Vallejos
Mamani
```

TUTOR: NAGIB LUIS VALLEJOS M.

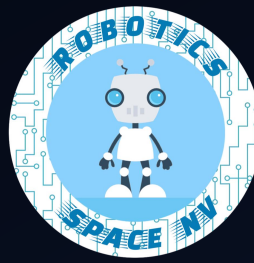


EJEMPLO 3 – CIRCUITO

Programar un pulsador en modo Switch para encender y apagar un sistema, imprimir “Encendido” cuando el sistema se ponga en funcionamiento e imprimir “Apagado” cuando el sistema se apague.



TUTOR: NAGIB LUIS VALLEJOS M.



EJEMPLO 3 – SOLUCIÓN

Programar un pulsador en modo Switch para encender y apagar un sistema, imprimir “Encendido” cuando el sistema se ponga en funcionamiento e imprimir “Apagado” cuando el sistema se apague.

S8-E3

```
1 int estado=0;
2 int estadoAnt=0;
3 int salida=0;
4 int led=3, pulsador=4;
5 void setup() {
6     pinMode(led, OUTPUT);
7     pinMode(pulsador, INPUT);
8     Serial.begin(9600);
9 }
10 void loop() {
11     estado=digitalRead(pulsador);
12     if(estado && estadoAnt==0) {
13         salida=1-salida;
14         delay(20);
15     }
16     estadoAnt=estado;
17     if(salida){
18         digitalWrite(led,1);
19         Serial.println("Sistema encendido");
20     }
21     else{
22         digitalWrite(led,0);
23         Serial.println("Sistema apagado");
24     }
25 }
```

TUTOR:NAGIB LUIS VALLEJOS M.



FUNCIÓN RANDOM()

Es una función integrada por defecto en arduino, la cual nos permite obtener números aleatorios.

Su sintaxis es la siguiente:

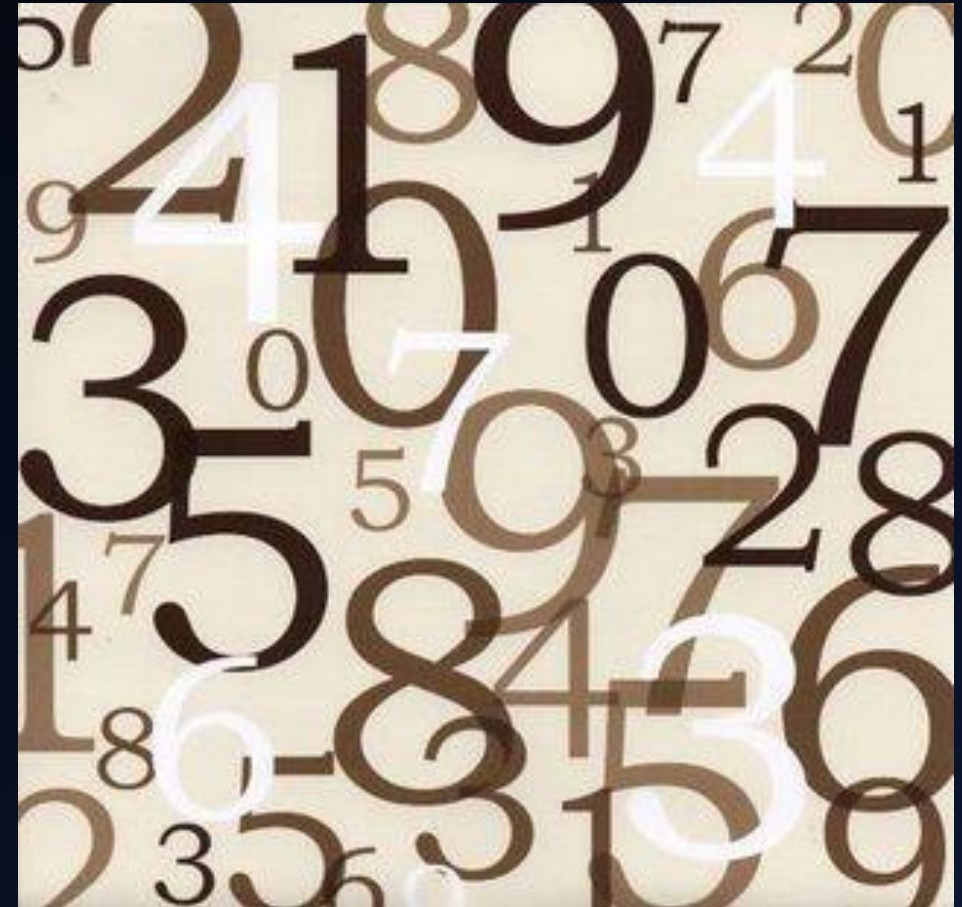
random(max);

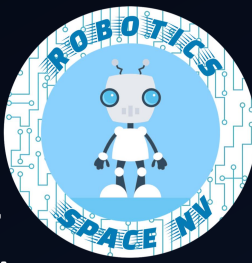
random(min,max);

Cabe mencionar que el valor máximo al que llegará es el valor asignado -1:

random(10); // <= 9

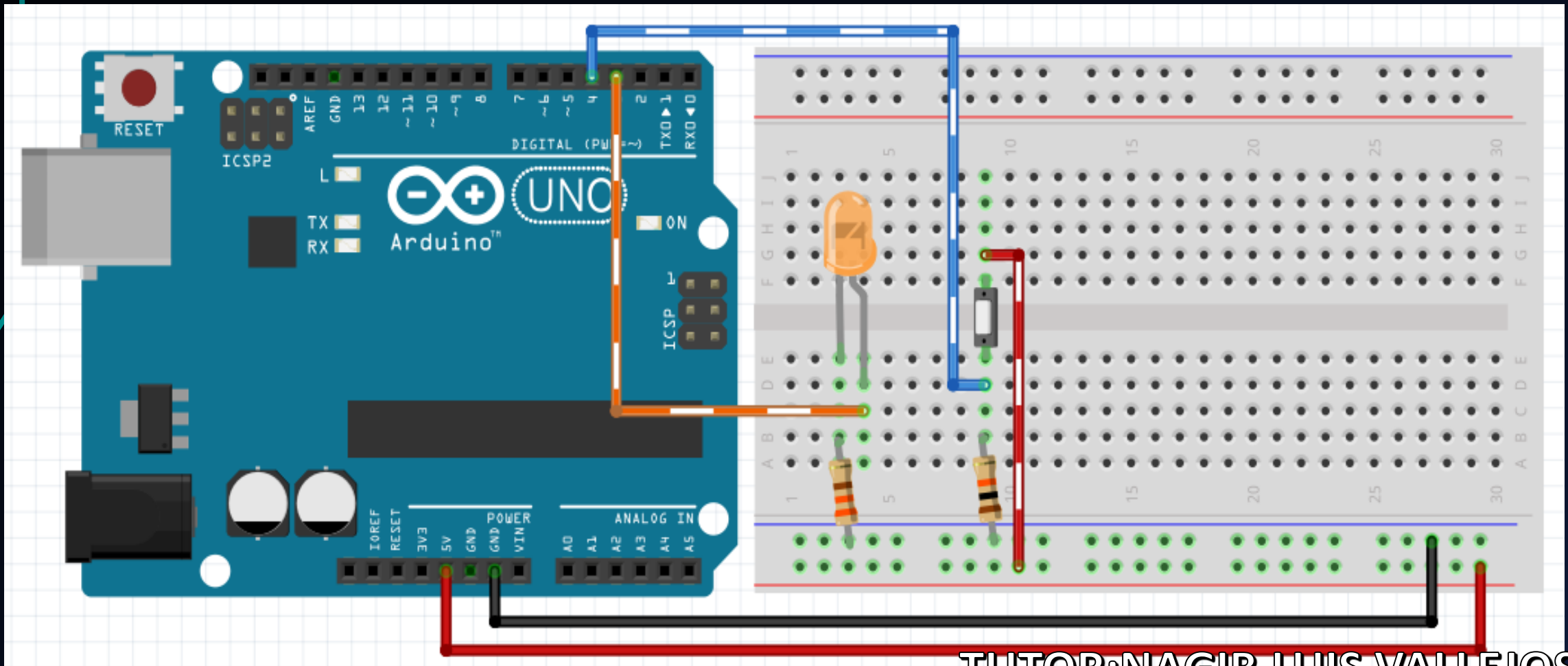
random(0,10); // <10



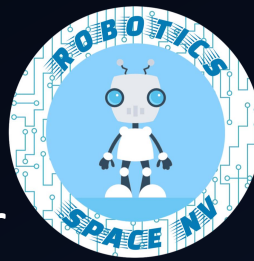


EJEMPLO 4 – CIRCUITO

Generar un número aleatorio entre 1 y 10, este número se debe imprimir por el monitor serie e indicar la cantidad de veces que se enciende y apaga un led a razón de 2 segundos, el número se activa tras presionar un pulsador.



TUTOR: NAGIB LUIS VALLEJOS M.



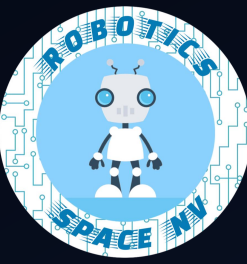
EJEMPLO 4 – SOLUCIÓN

Generar un número aleatorio entre 1 y 10, este número se debe imprimir por el monitor serie e indicar la cantidad de veces que se enciende y apaga un led a razón de 2 segundos, el número se activa tras presionar un pulsador.

S8-E4

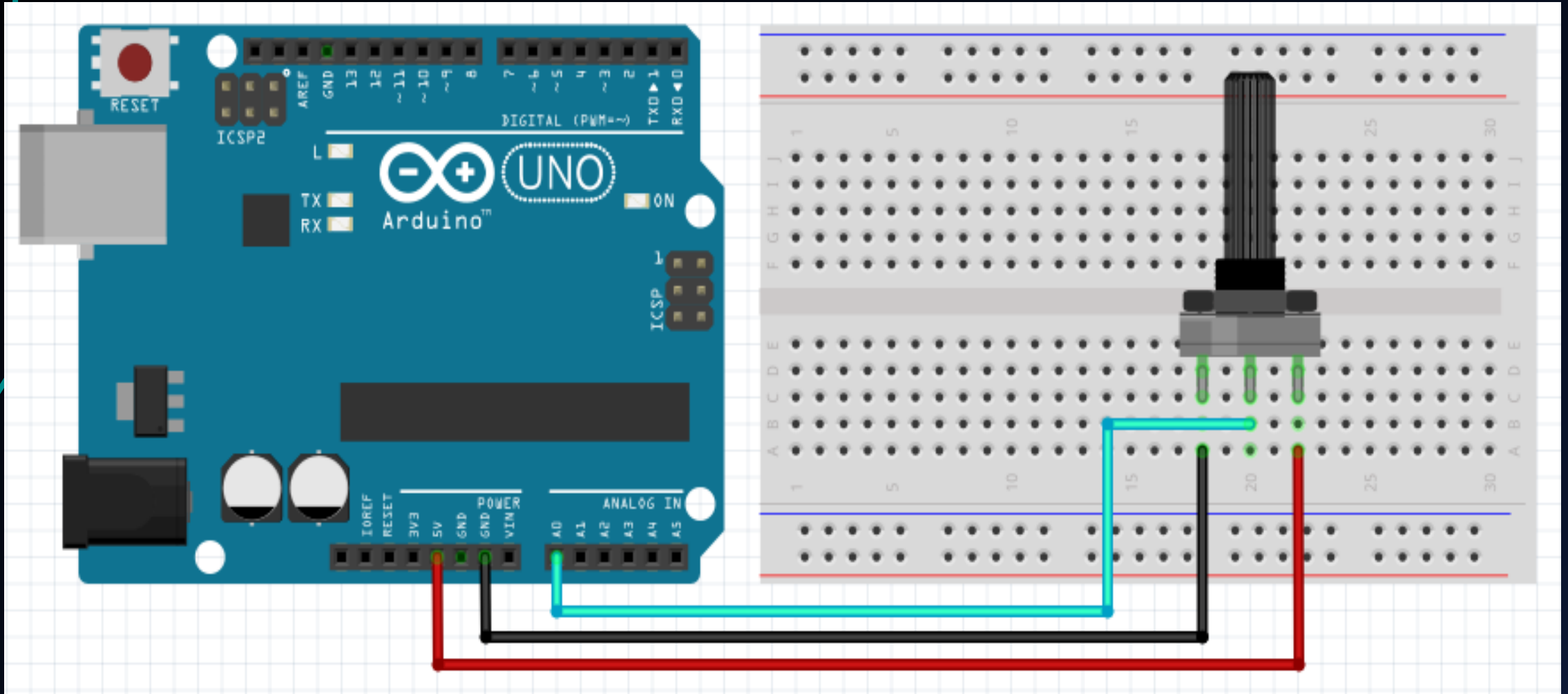
```
1 int led=3, pulsador=4, estado;
2 int aleatorio;
3 void setup() {
4     pinMode(led, OUTPUT);
5     pinMode(pulsador, INPUT);
6     Serial.begin(9600);
7 }
8
9 void loop() {
10     estado=digitalRead(pulsador);
11     if(estado){
12         aleatorio=random(1,10);
13         Serial.println(aleatorio);
14         for(int i=0; i<aleatorio;i++){
15             digitalWrite(led,1);
16             delay(250);
17             digitalWrite(led,0);
18             delay(250);
19         }
20     }
21 }
```

TUTOR:NAGIB LUIS VALLEJOS M.

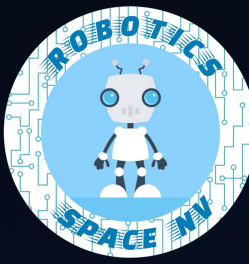


EJEMPLO 5 – CIRCUITO

Imprimir el valor de lectura de un potenciómetro a razón de 500 ms



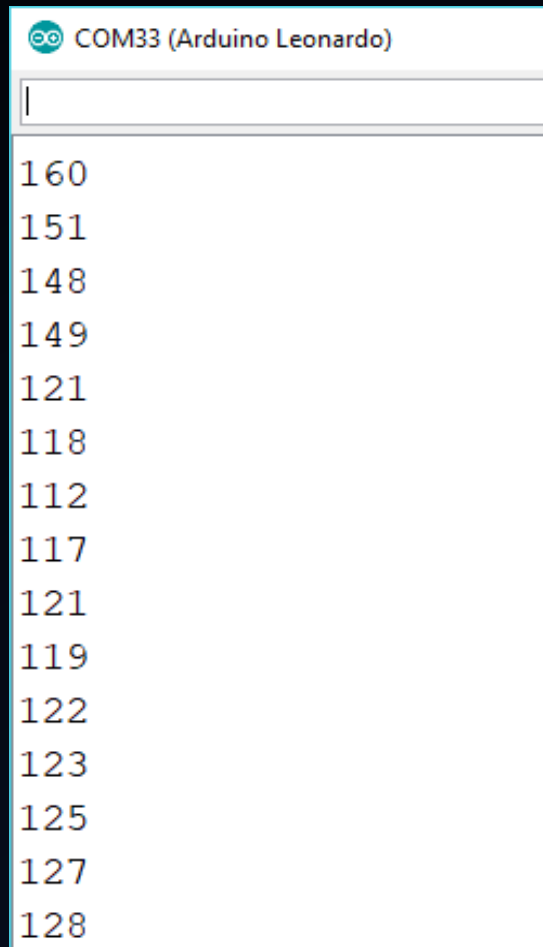
TUTOR:NAGIB LUIS VALLEJOS M.



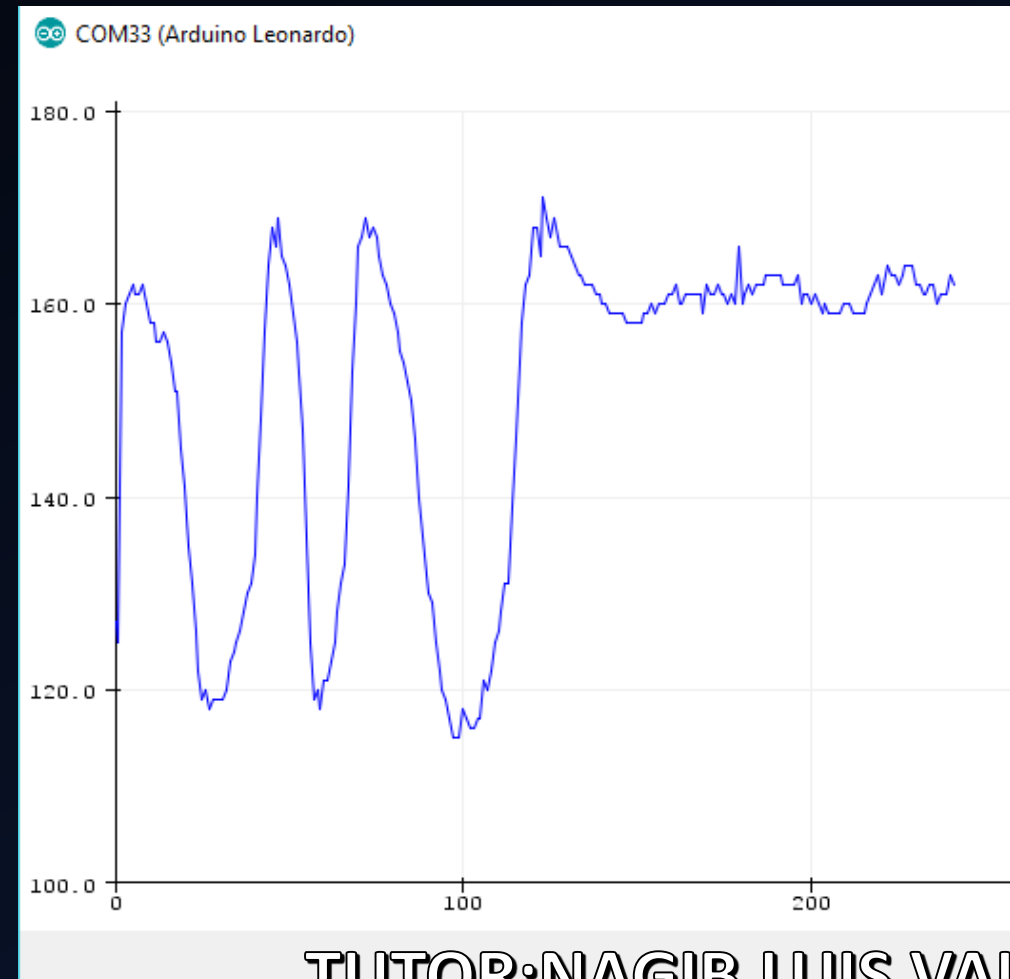
EJEMPLO 5 – PRUEBAS

Imprimir el valor de lectura de un potenciómetro a razón de 500 ms

MONITOR SERIAL



SERIAL PLOTTER



TUTOR: NAGIB LUIS VALLEJOS M.



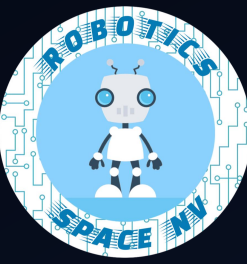
EJEMPLO 5 – SOLUCIÓN

Imprimir el valor de lectura de un potenciómetro a razón de 500 ms

S8-E5

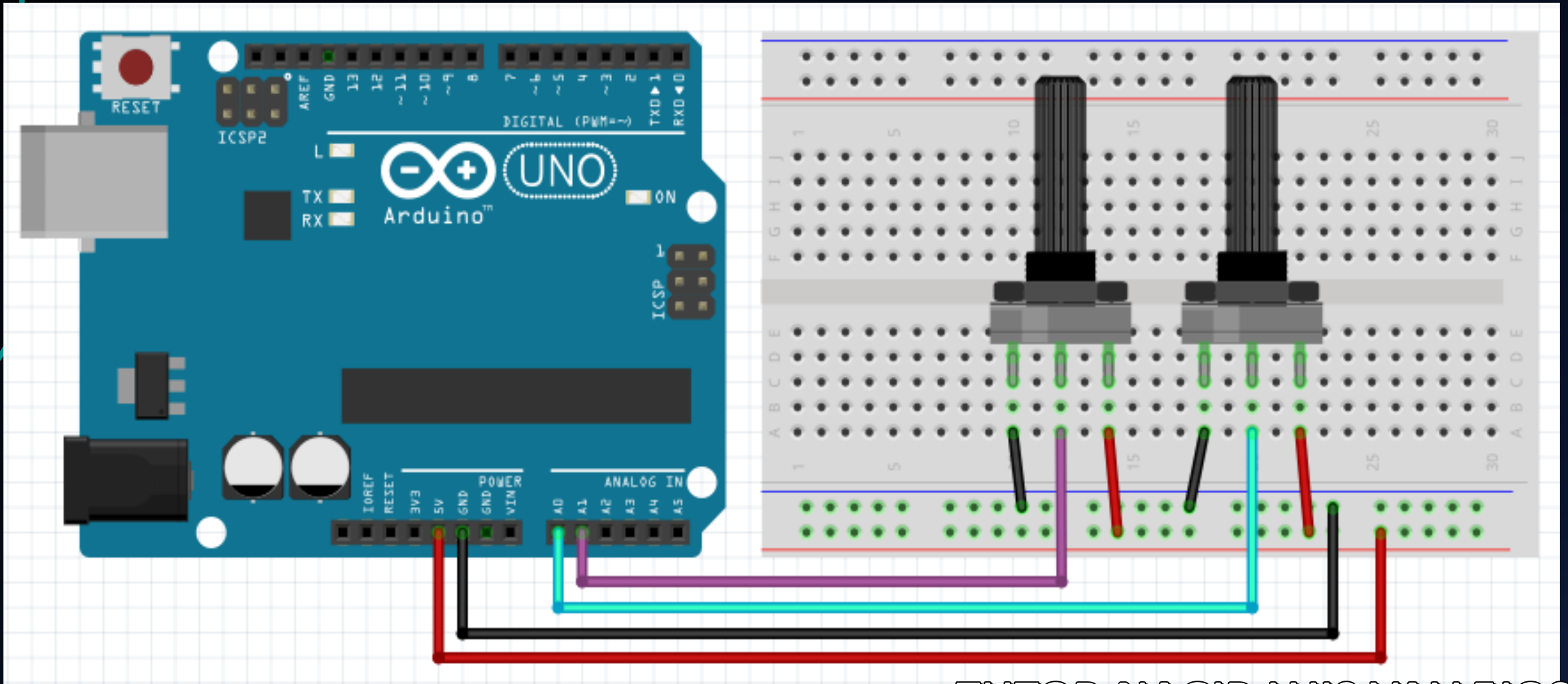
```
1 int potenciometro=A0,lectura;
2 void setup() {
3     Serial.begin(9600);
4 }
5 void loop() {
6     lectura=analogRead(potenciometro);
7     Serial.println(lectura);
8     delay(500);
9 }
```

TUTOR:NAGIB LUIS VALLEJOS M.

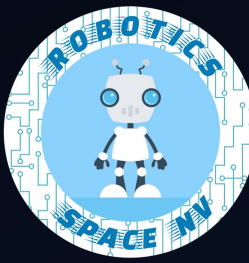


EJEMPLO 6 – CIRCUITO

Graficar el valor de 2 potenciómetros de manera paralela a través del serial plotter a razón de 500 ms

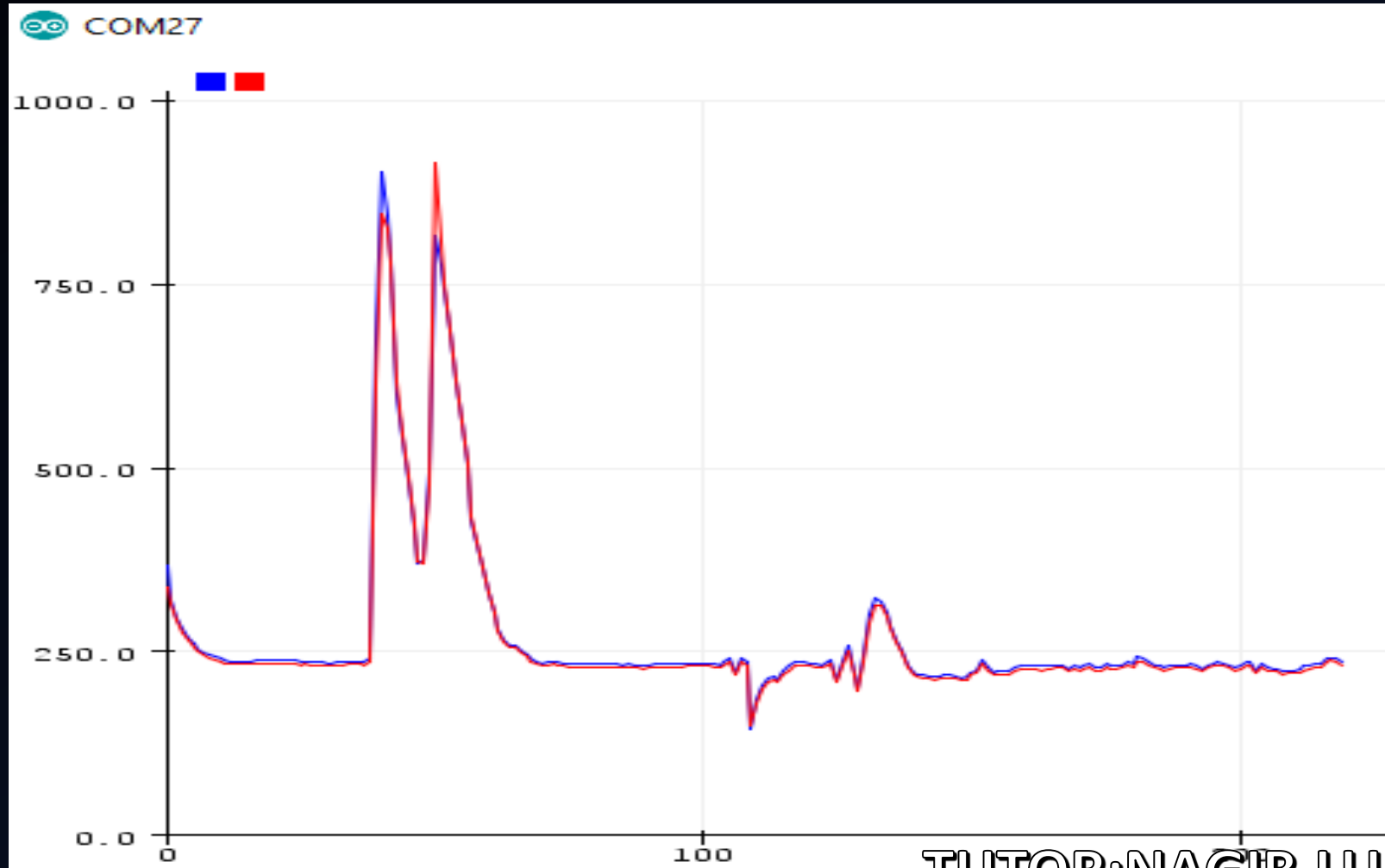


TUTOR:NAGIB LUIS VALLEJOS M.

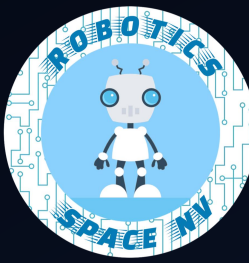


EJEMPLO 6 – PRUEBAS

Graficar el valor de 2 potenciómetros de manera paralela a través del serial plotter a razón de 500 ms



TUTOR: NAGIB LUIS VALLEJOS M.



EJEMPLO 6 – SOLUCIÓN

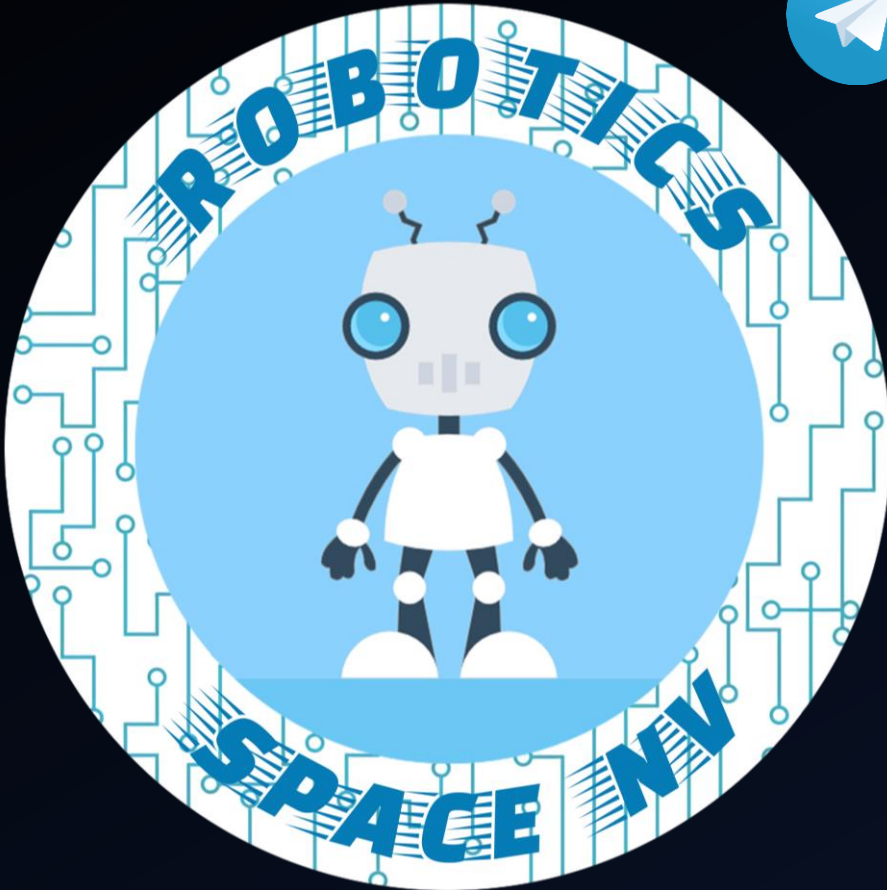
Graficar el valor de 2 potenciómetros de manera paralela a través del serial plotter a razón de 500 ms

S8-E6

```
1 int potenciometro=A0,potenciometro2=A1,lectura,lectura2;
2 void setup() {
3     Serial.begin(9600); //inicializa el monitor
4 }
5
6 void loop() {
7     lectura=analogRead(potenciometro);
8     lectura2=analogRead(potenciometro2);
9     Serial.print(lectura);
10    Serial.print(",");
11    Serial.println(lectura2);
12    delay(500);
13 }
```

TUTOR:NAGIB LUIS VALLEJOS M.

CONTACTOS



(+591) 63096640



robotics.space.nv@gmail.com



fb.me/RoboticsSpaceNV



@NagibVallejos



Robotics Space NV



<https://github.com/nagibvalejos/Robotics-Space-NV>

TUTOR:NAGIB LUIS VALLEJOS M.