Day 2:
Controlar los partes

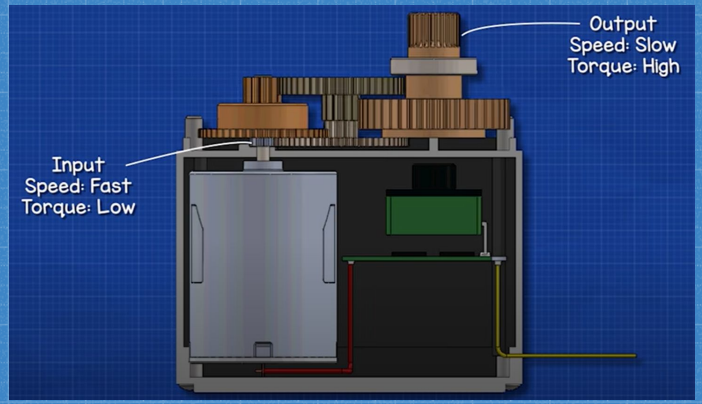
Necesitas:

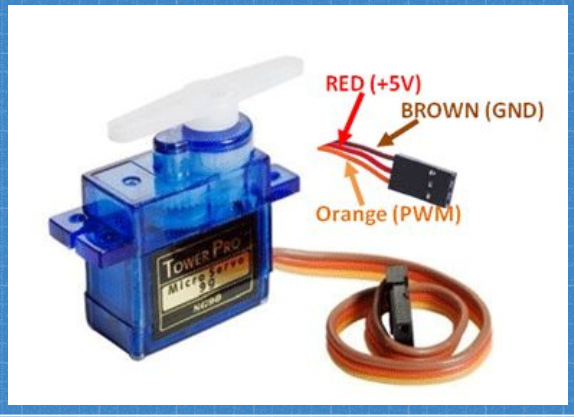
- Servo Motor
- Boton
- 1K Resistor

1

Servo Motor

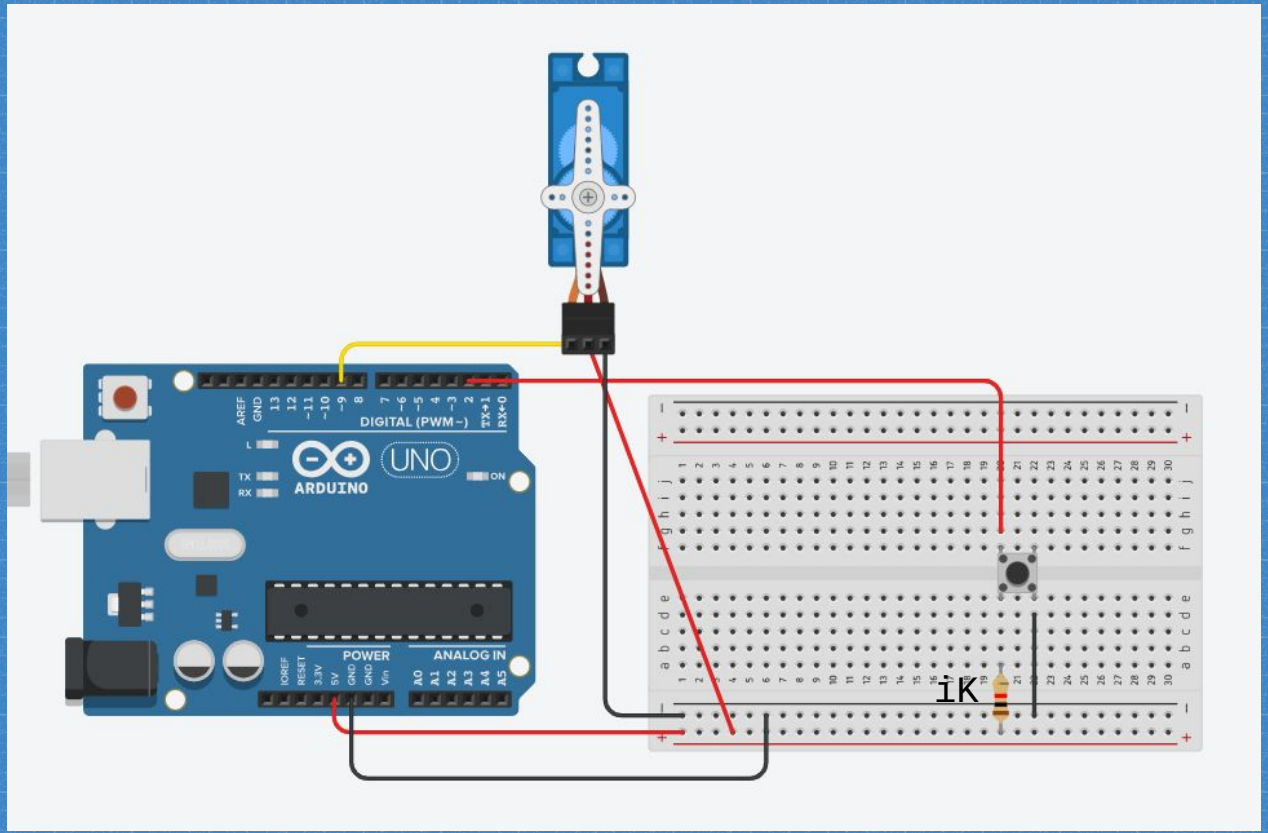
Cómo Funciona





Ejemplo de aplicación:
brazo robótico

Controlar un servo (30 grados, 180 grados)



Nuevos métodos:

- .attach()
- .write()
- Serial.println()

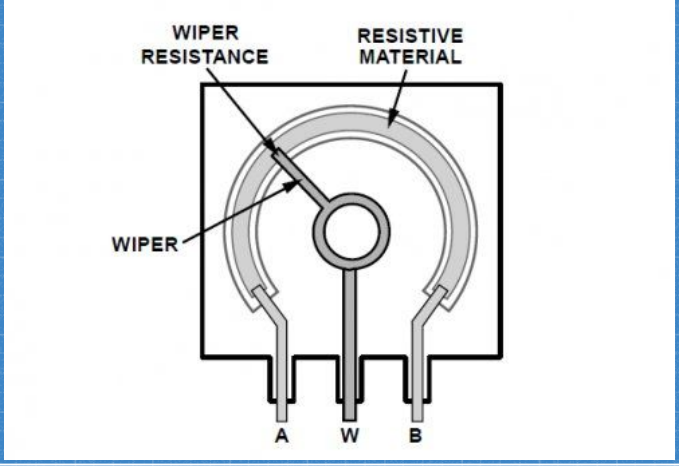
Necesitas:

- Palanca de mando
- Servo Motor
- Motor de DC

2
Palanca de mando

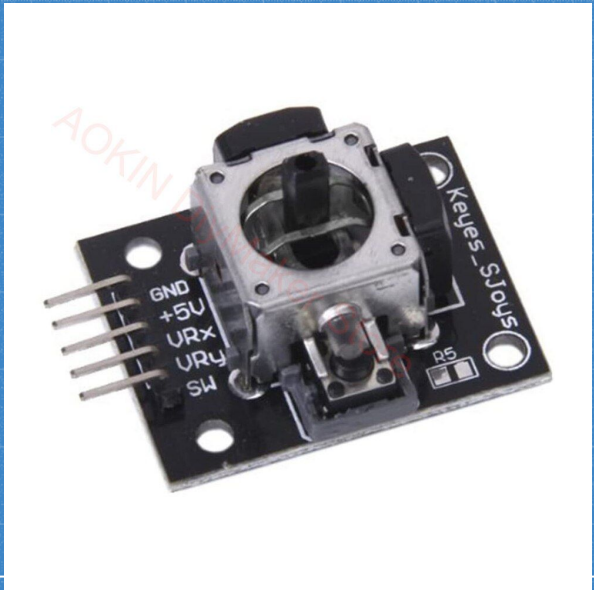
Cómo Funciona

- Potenciómetro = resistencia variable
- salida analógica (0-1023)
  - Usar: analog pins, analogRead()



Cómo Funciona

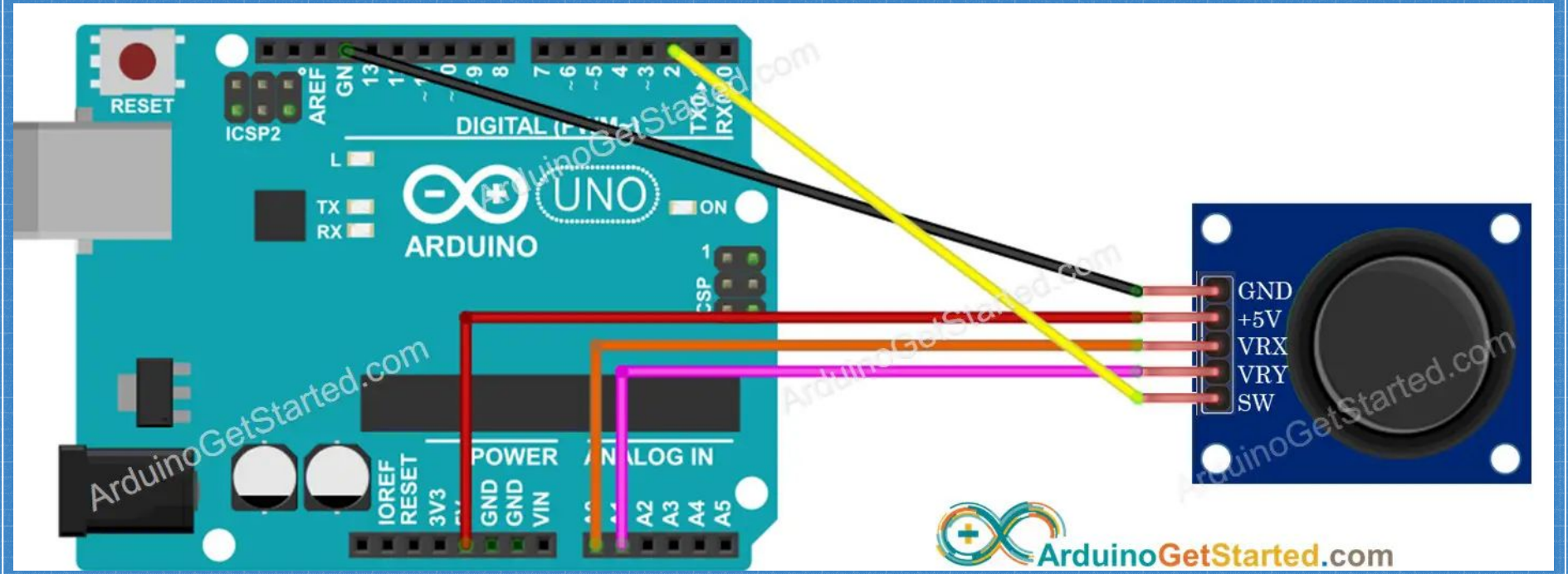
- 2 potenciómetros (x, y) y un botón
- VRx = movimiento horizontal, VRy = movimiento vertical



Serial print: x, y

Nuevos métodos:

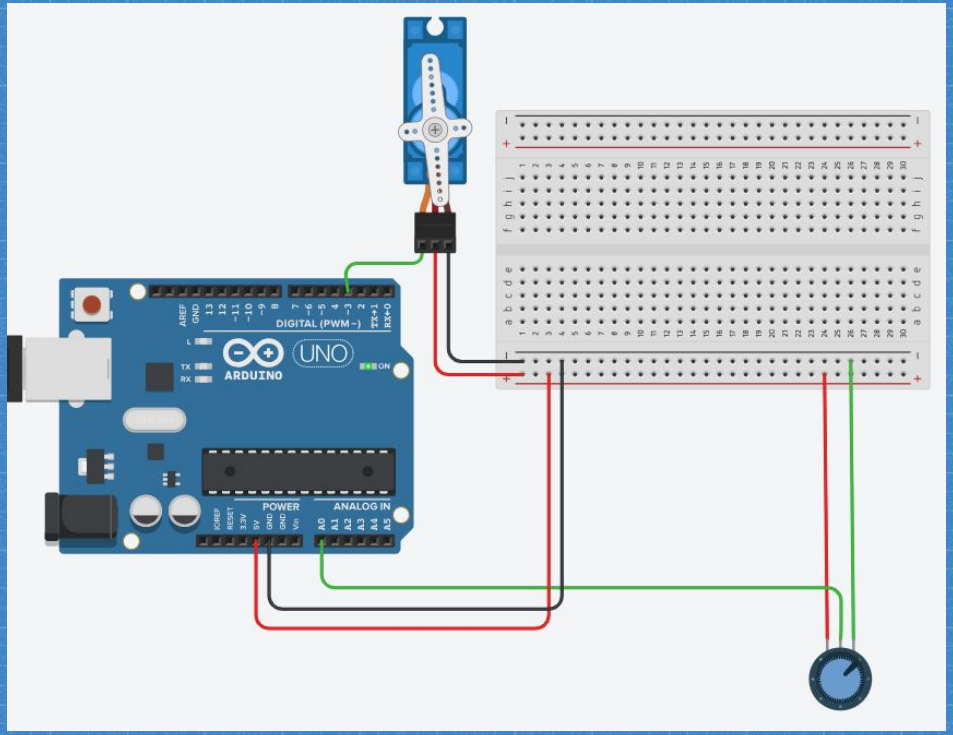
- analogRead()

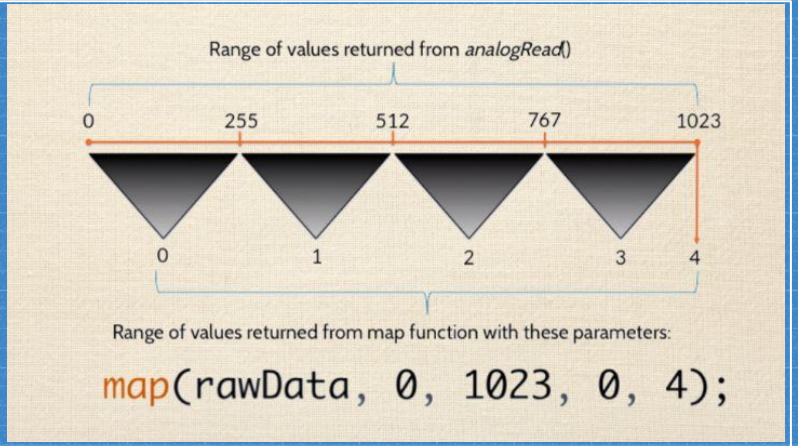


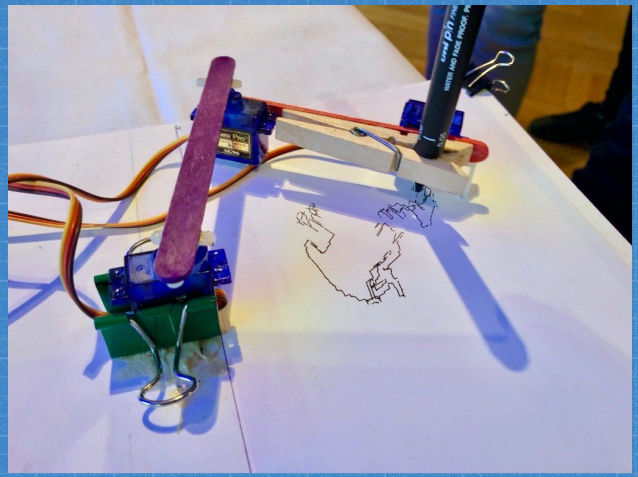
Controlar un motor de Servo

Nuevos métodos:

- map()

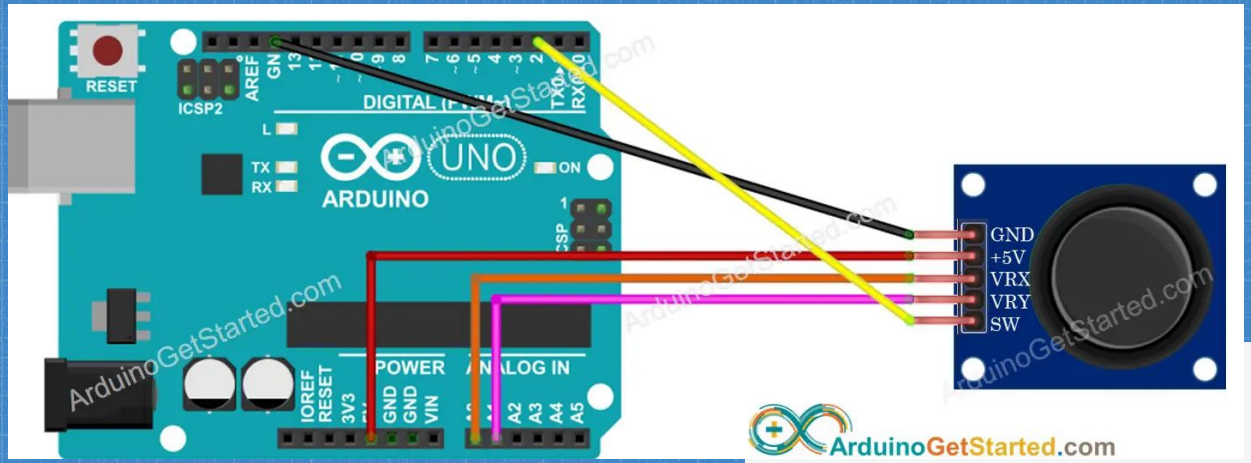




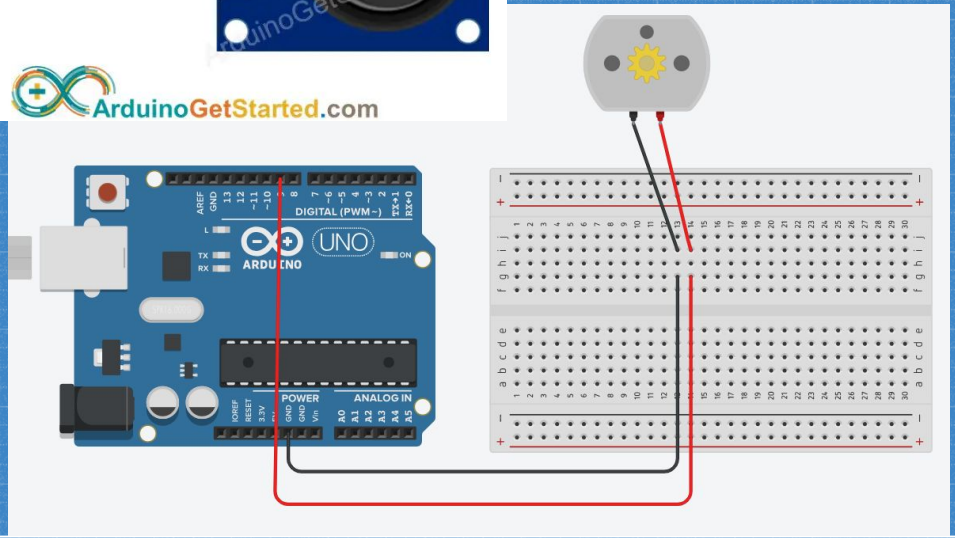


Ejemplo de aplicación: controlar múltiples servos

Controlar un motor de DC



+

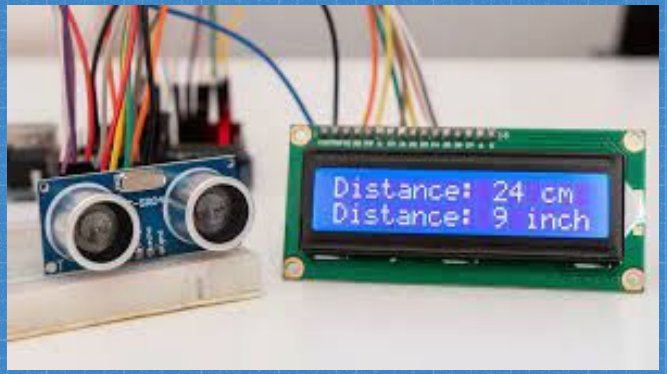


Necesitas:

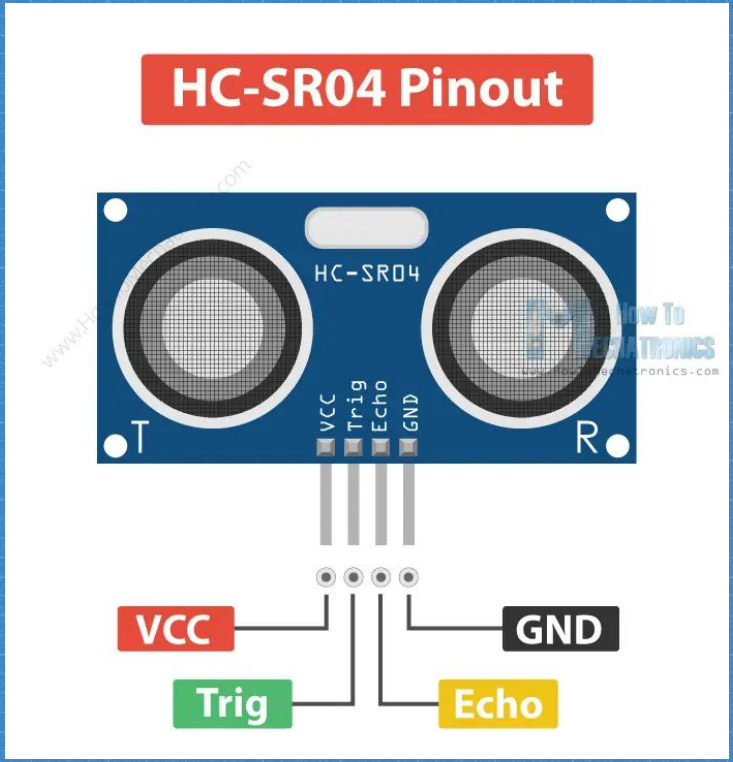
- Sensor Ultrasónico
- LCD
- Resistor: 220Ω, 1kΩ

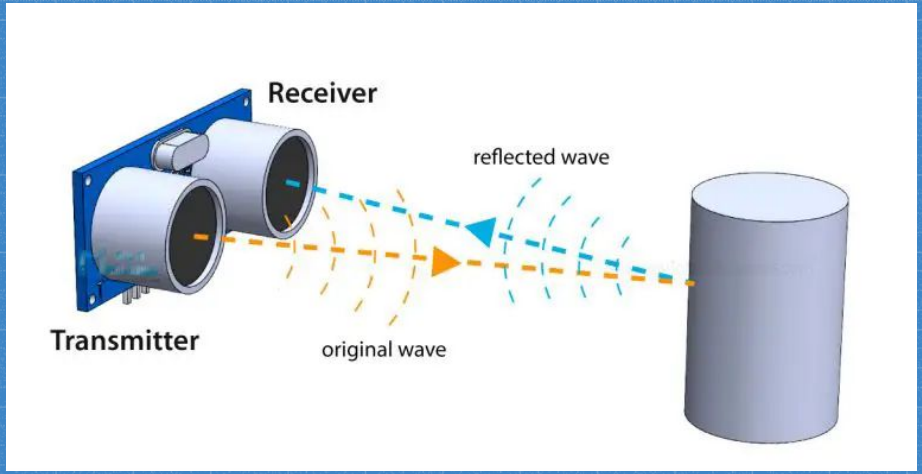
3

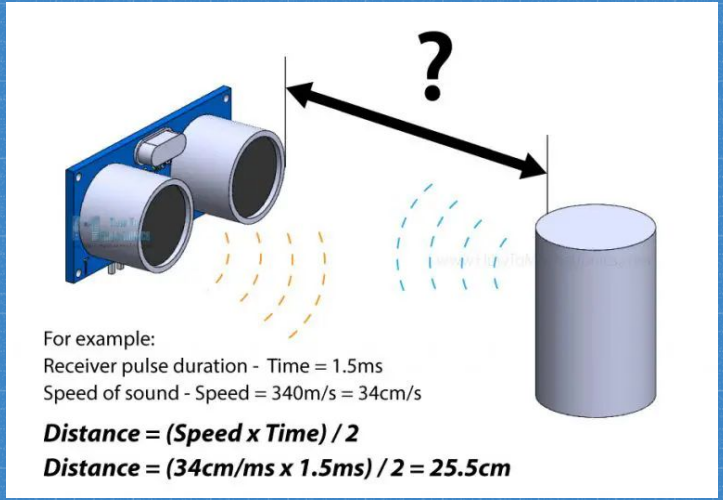
Sensor Ultrasónico/LCD



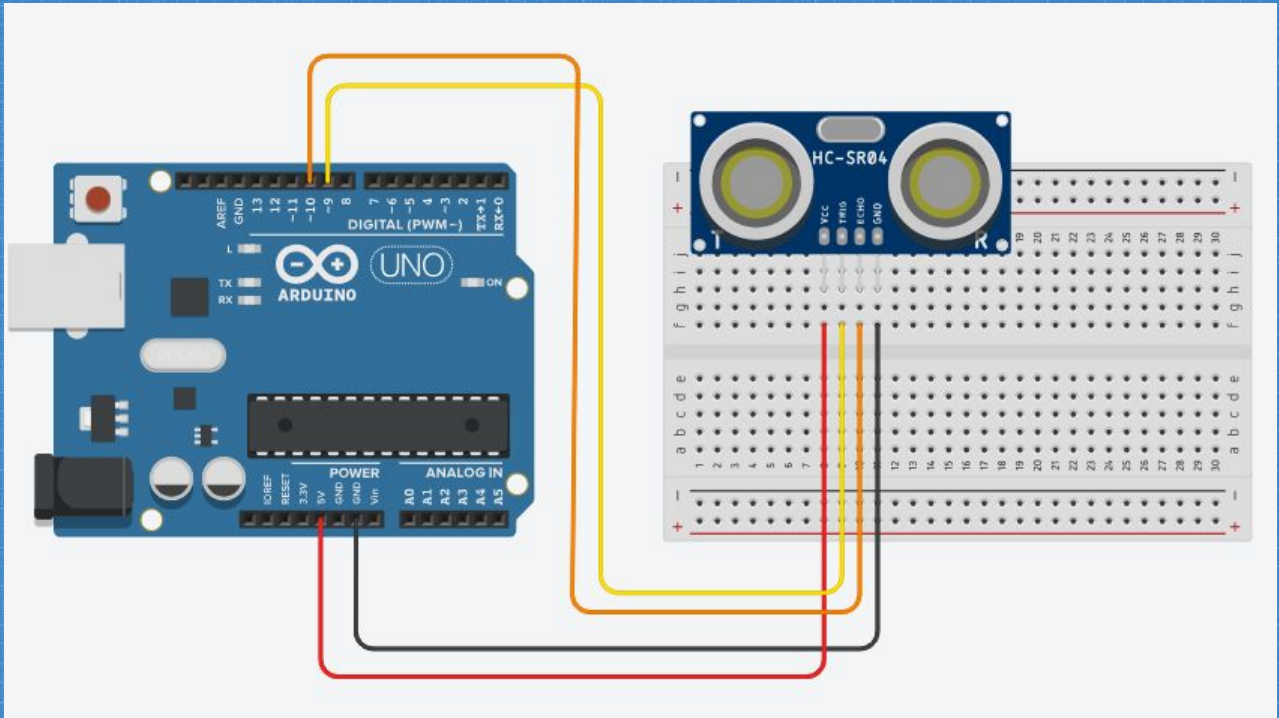
Cómo Funciona







Mostrar la distancia - Serial Monitor

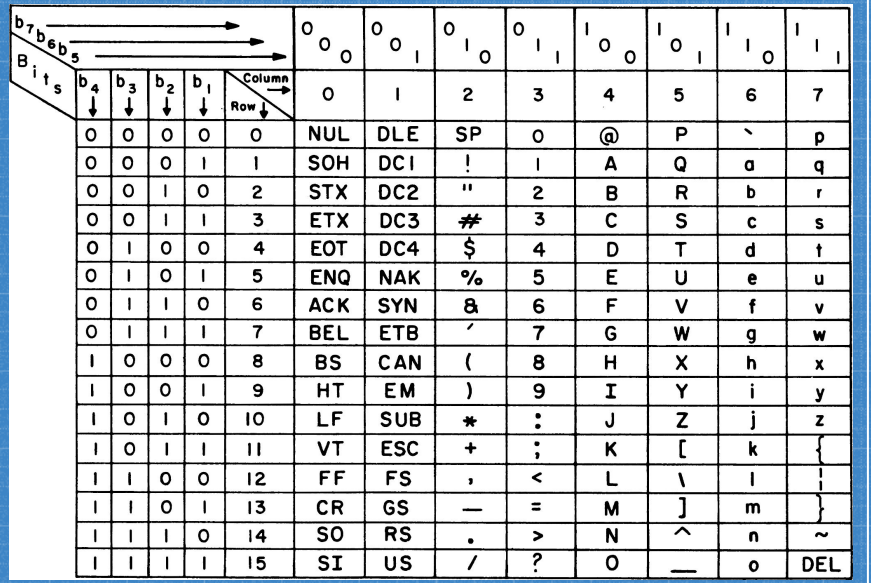


Nuevos métodos:

- delayMicroseconds()
- pulseIn()

Mostrar la distancia - LCD

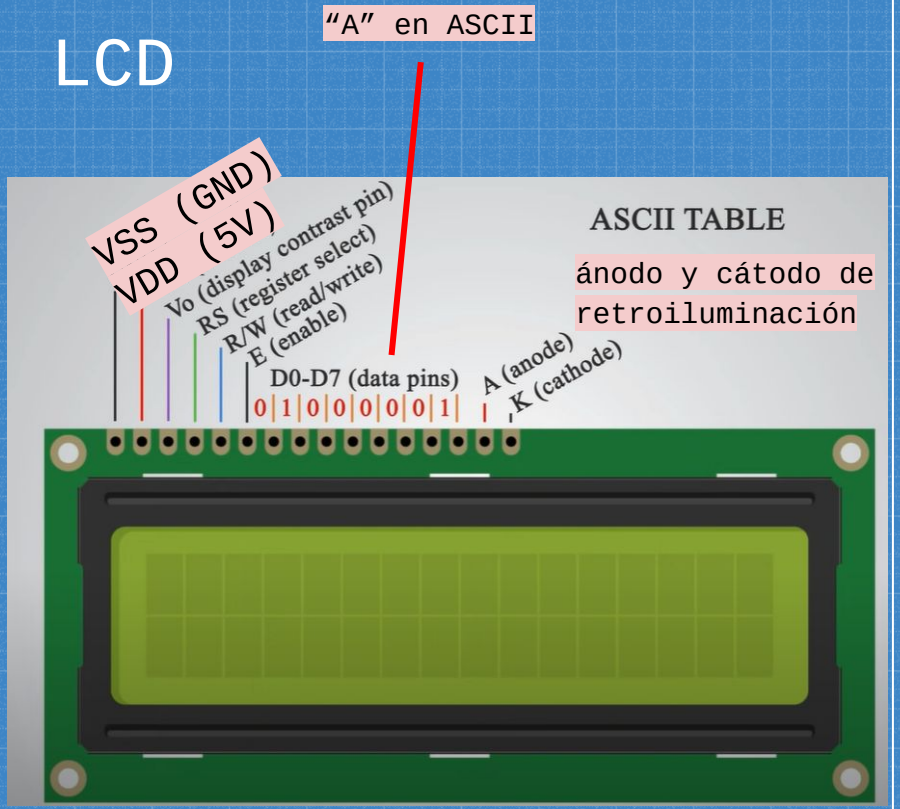
- Cada símbolo tiene un representación binario (ASCII)
- 1 ‘bit’ es un dígito binario y 8 bits son un ‘byte’



Bits b_4	b_3	b_2	b_1	Row / Column	0 (000)	1 (001)	2 (010)	3 (011)	4 (100)	5 (101)	6 (110)	7 (111)
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	|
1	1	0	1	13	CR	GS	—	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL

Mostrar la distancia - LCD

- **D0-D7**: Recibe los datos (8 bits)
- **Vo**: controlar el contraste
- **RS**: Transmitir ‘datos’ (los caracteres) o ‘mandatos’ (e.x: ‘setCursor()’)
- **R/W**: Read (Mandar datos a la computadora), Write (Mandar datos al LCD)
- **E**: permite escribir en los siguientes 8 pines

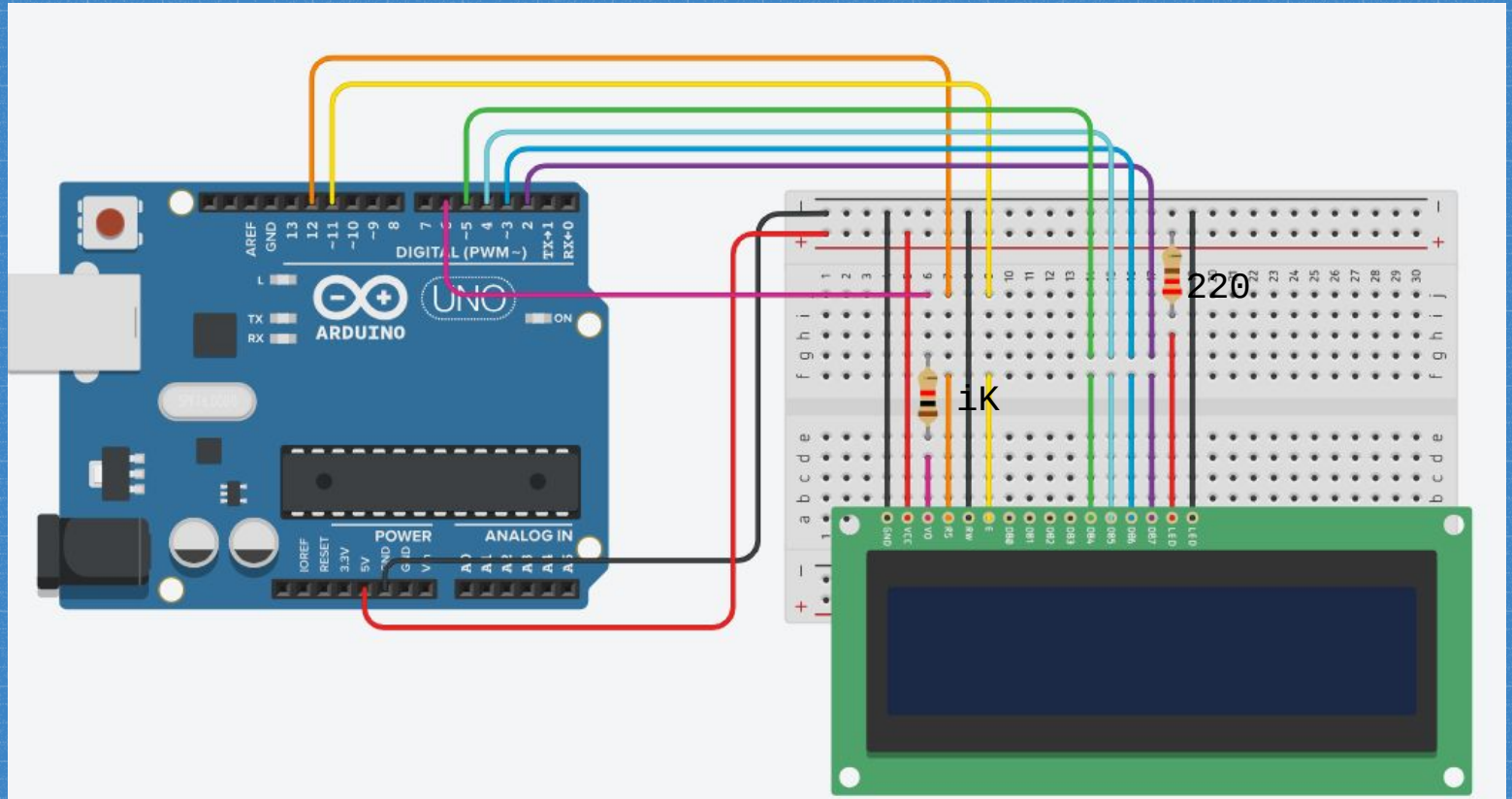


Opciónal: D0-D3

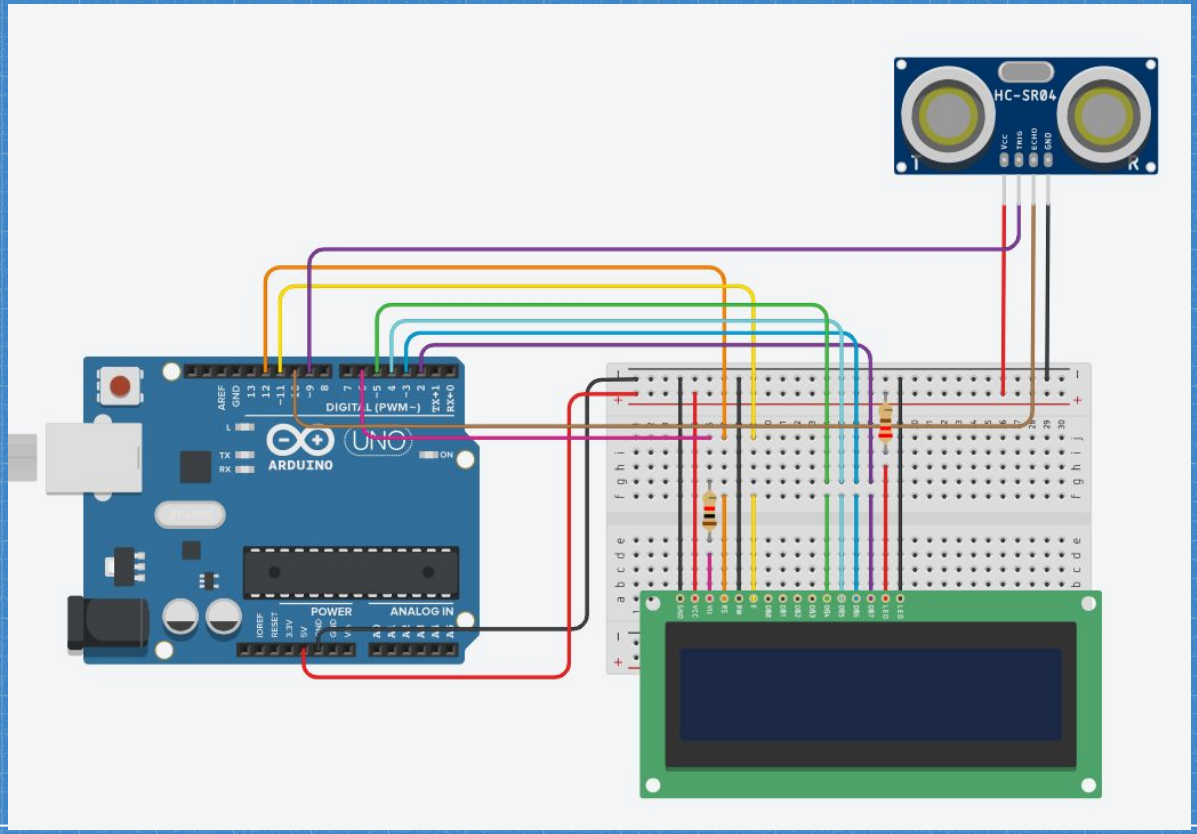
Mostrar la distancia - LCD

Nuevos métodos:

- .begin()
- .setCursor()
- .print()



Mostrar la distancia - LCD



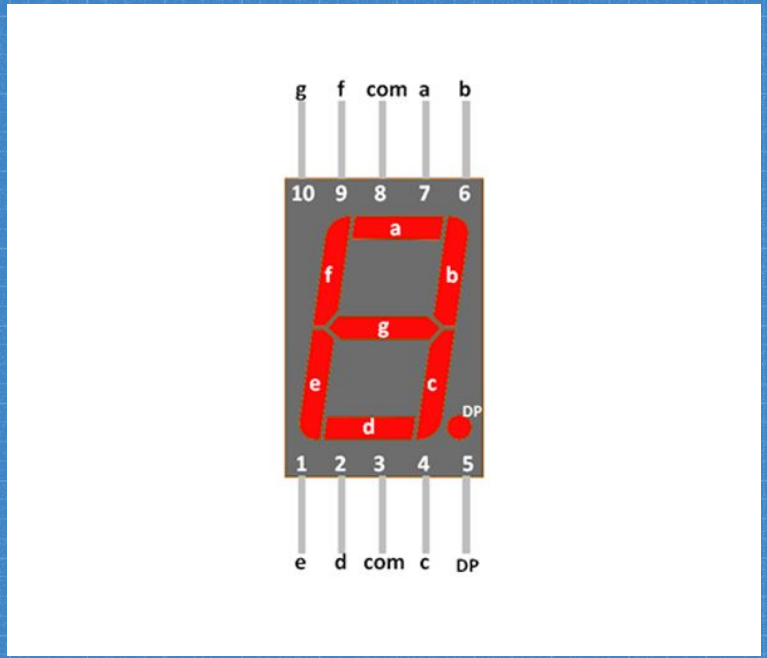
Nuevos métodos:

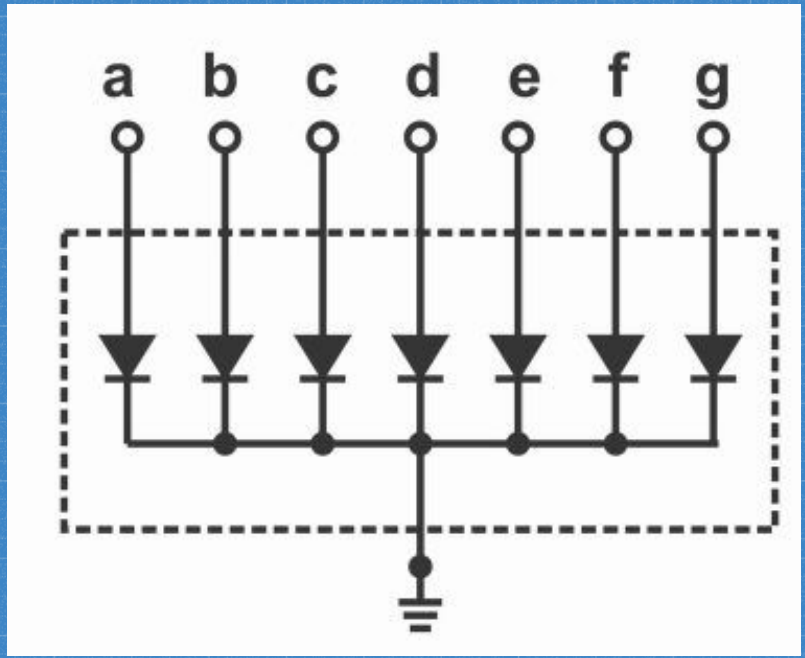
- .clear()

4

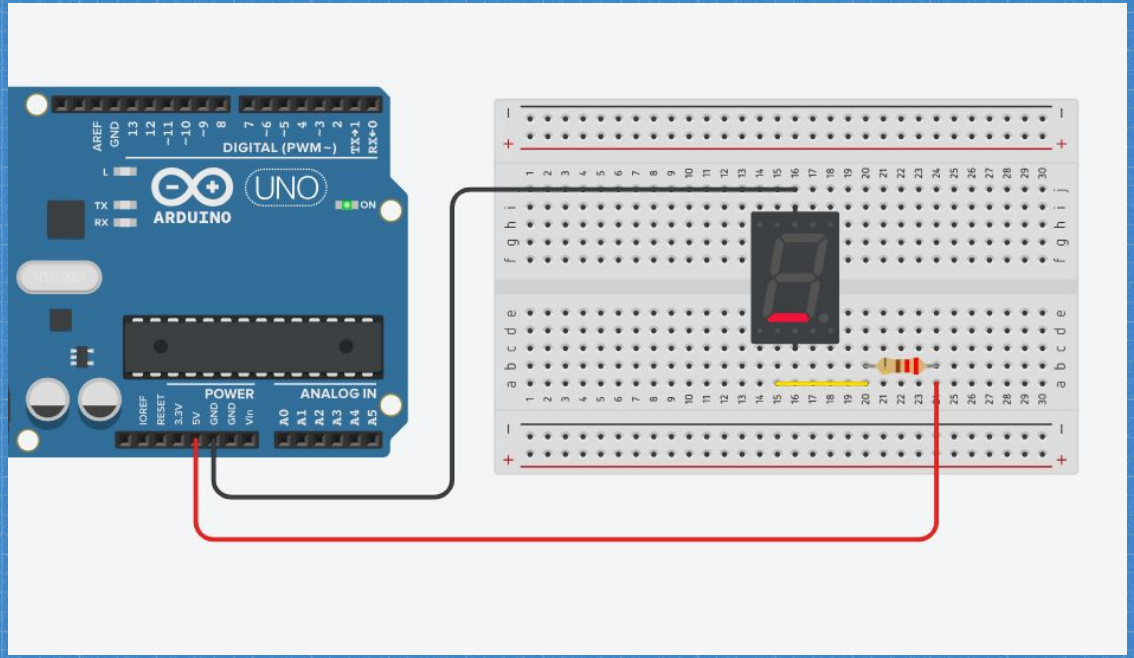
Pantalla de 1 dígito

Cómo Funciona





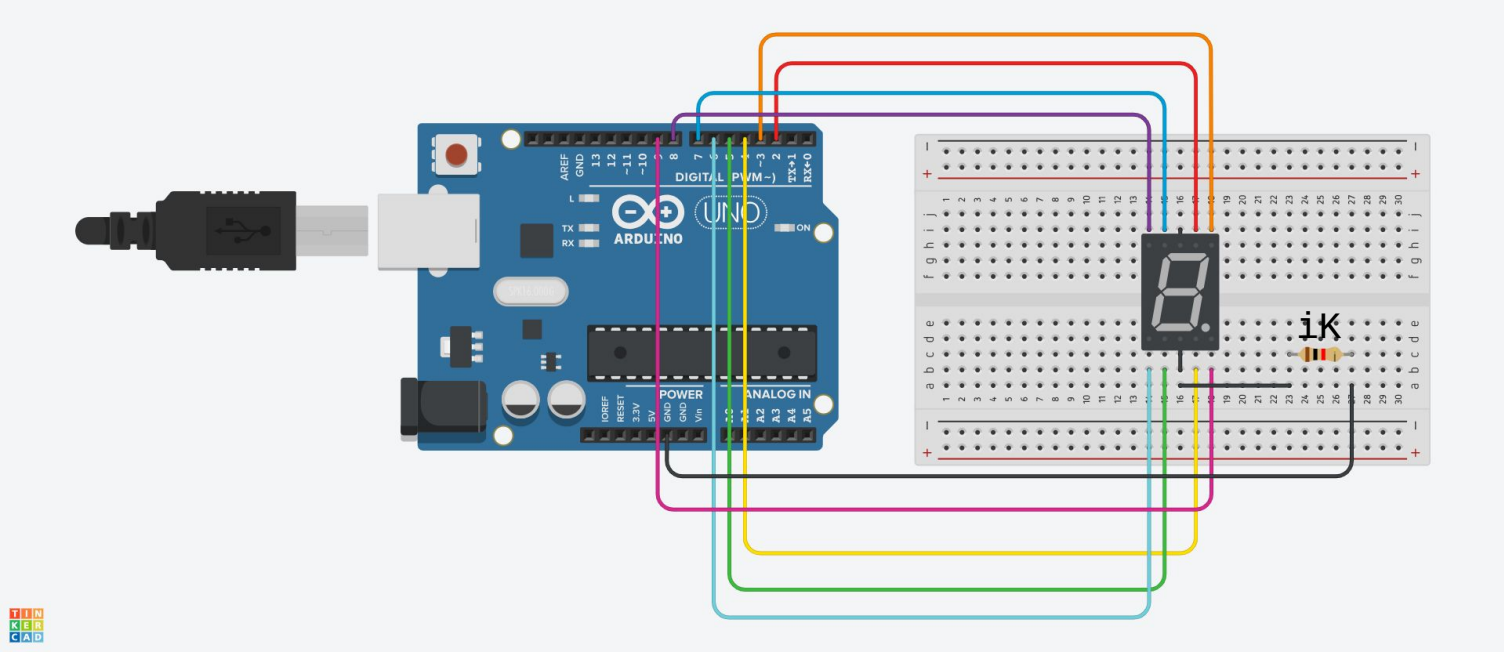
Demo



Schematics

Nuevos métodos:

- .begin()
- .setBrightness()
- setNumber()
- .refreshDisplay()



Coding - Python