

18 Taller 3: Reto # 1

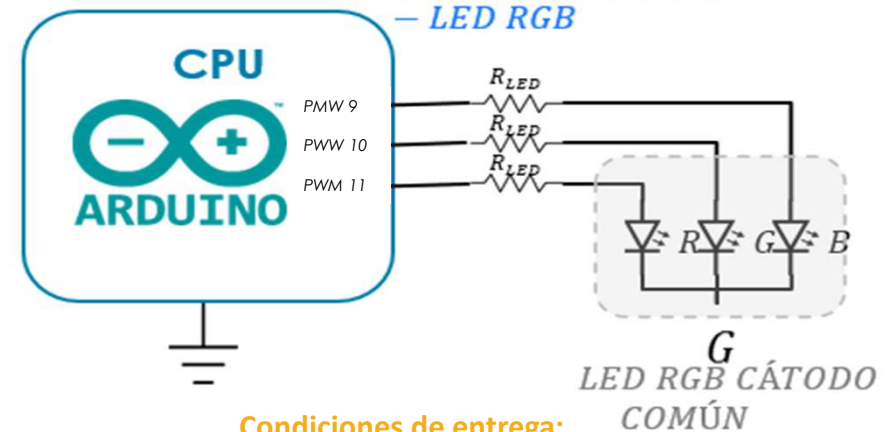
Diseñe un código en Arduino para controlar un LED RGB a través de un teclado matricial. Con la tecla '*' seleccionar los canales para modificar el valor del color en dicho canal, con la tecla '#' para confirmar los cambios.

Materiales:

- Arduino
- LED RGB
- Teclado Telefónico
- Display LCD 16*2 I2C

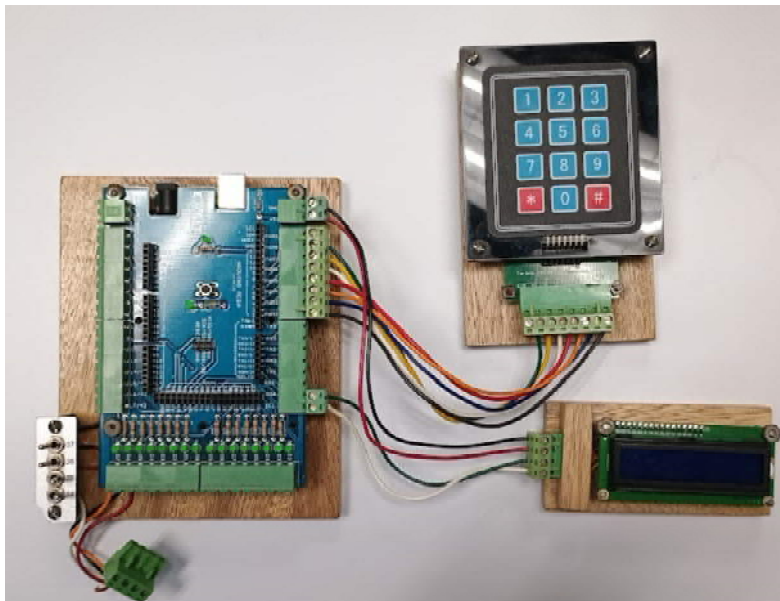
C	O	N	T	R	O	L		L	E	D		R	G	B	
	R	2	5	5		G	2	5	5		B	2	5	5	

*Circuito típico PWM
– LED RGB*



Condiciones de entrega:

- Enviar y adjuntar los códigos en interactiva.
- Realizar un video explicando el funcionamiento.
- Evitar copia de códigos, ya que se anula la entrega a los grupos involucrados, también, si el código esta realizado en Chat GPT.



Tips: investigar el método `isDigit(key)` y convertir una variable tipo `char` a entero. Ej: `'1'-48 = 1`.

19 Taller 3: Reto # 2

Diseñe un código en Arduino para controlar un motor DC a través de un teclado matricial. Con la tecla '*' seleccionar para modificar la velocidad, con la tecla '#' para confirmar los cambios. En ese mismo orden, con la tecla '*' cambiar invertir el sentido de giro y detener el motor (crear un arreglo para controlar el motor. Ej: `int control[3] = {0x00, 0x01, 0x02};`).

Materiales:

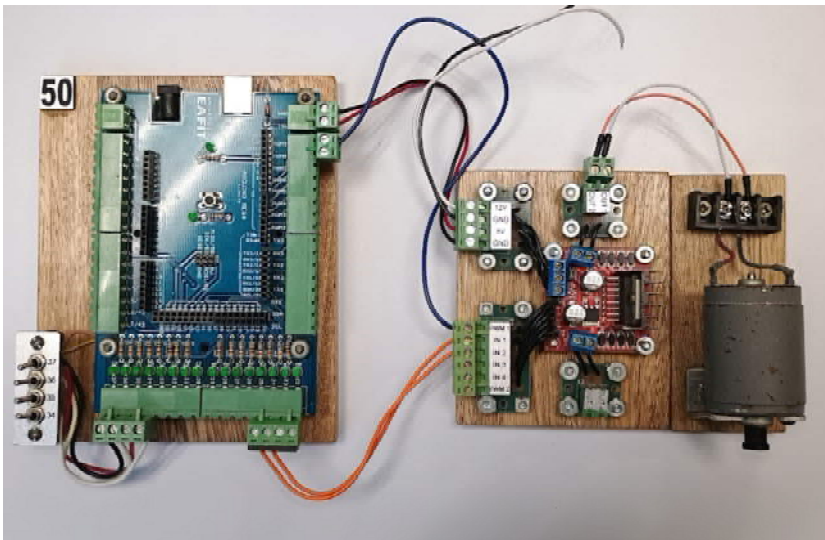
- Motor DC + Puente H + Fuente 12vdc
- Arduino
- Teclado telefónico
- Display LCD 16 * 2 - I2C

C	O	N	T	R	O	L		M	O	T	O	R		D	C
S		1	0	0	%		R		O	N		L		O	N

Condiciones de entrega:

- Enviar y adjuntar los códigos en interactiva.
- Realizar un video explicando el funcionamiento.
- Evitar copia de códigos, ya que se anula la entrega a los grupos involucrados, también, si el código esta realizado en Chat GPT.

Tips: investigar el método `isDigit(key)`, `map(valor, 0, 255, 0, 100)` y convertir una variable tipo char a entero. Ej: `'1' - 48 = 1`.



20 Taller 3: Reto # 3

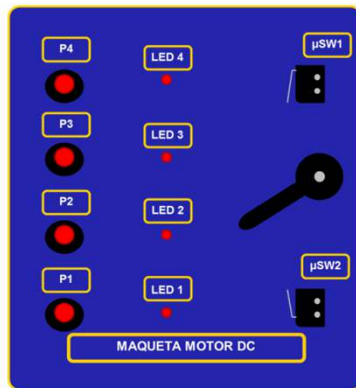
Diseñe un código en Arduino para simular en llenado de un tanque de agua a través de la maqueta MOTOR DC. Con la tecla '*' se programa el llenado, con el pulsador 'PX' se inicia el proceso de llenado. En ese mismo orden, la tecla '#' se programa para vaciar el tanque. El vaciado finaliza cuando la plumilla hace contacto con uSW2.

Materiales:

- Maqueta MOTOR DC Arduino
- Arduino
- Teclado telefónico
- Display LCD 16 *2 - I2C

MOTOR DC ARDUINO ELECTRÓNICA

Identificación de Entradas	
PUERTO	FUNCION
37	Microsuiche arriba(uSW1)
36	Microsuiche abajo (uSW2)
35	Pulsador P1
34	Pulsador P2
33	Pulsador P3
32	Pulsador P4
Identificación de Salidas	
PUERTO	FUNCION
22	Subir Motor
23	Bajar Motor
24	Led 1
25	Led 2
26	Led 3
27	Led 4



Condiciones de entrega:

- Enviar y adjuntar los códigos en interactiva.
- **Todo el equipo de trabajo debe presentar la entrega explicando el funcionamiento.**
- Evitar copia de códigos, ya que se anula la entrega a los grupos involucrados, también, si el código esta realizado en Chat GPT.

C	O	N	T	R	O	L		D	E		N	I	V	E	L
		N	I	V	E	L			1	0	0	%			

Pasos:

- Cuando inicie el programa la plumilla o indicador de nivel análogo debe estar abajo (haciendo contacto con uSW2) y los Leds 1, 2, 3 y 4 deben estar apagados.
- Para programar los procesos de llenado el tanque debe estar vacío.
- Por teclado programar los siguientes estados de llenado:
 1. El llenado máximo se realiza ingresando por teclado el nivel requerido. Si Nivel == 100 y P4 ==1, el motor de la plumilla se activa hasta leer el uSW1. (Encender los Leds 1, 2, 3 y 4).
 2. Programar el llenado a 75%. Si Nivel == 75 y P3==1, la plumilla se debe detener aproximadamente donde dice LED 3 de la maqueta. . (Encender los Leds 1, 2, 3 y apagar el 4) . Deben usar la función `millis()` para estimar el tiempo desplazamiento.
 3. Programar el llenado a 50%. Si Nivel == 50 y (P2==1 o P3==1), la plumilla se debe detener aproximadamente entre LED 3 y LED 2 de la maqueta. . (Encender los Leds 1 y 2, y apagar 3 y 4) . Deben usar la función `millis()` para estimar el tiempo desplazamiento.
 4. Programar el vaciado del tanque con la tecla '*' y el pulsador P1, la plumilla debe bajar, y se debe detener hasta leer el uSW2. Apagar todos los leds

Tips: investigar el método `isDigit(key)`, `map(valor, 0, 255, 0, 100)` y convertir una variable tipo char a entero. Ej: '1'-48 = 1.

