

**Parte 3**

# Lección

# 3

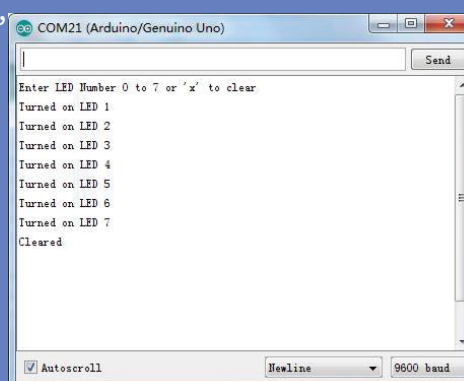
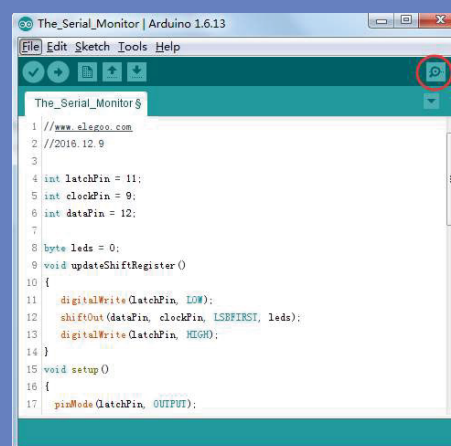
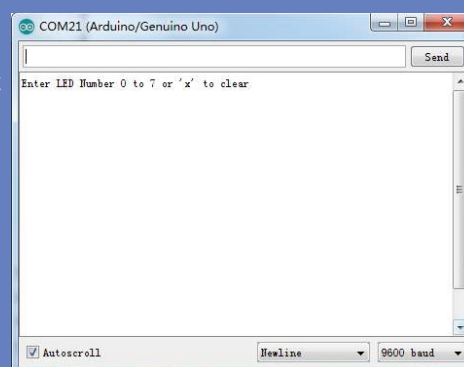
**El monitor Serial**

## Resumen

En este curso, añadirá la lección 15 de la parte 2 para el control Computadora que utiliza el monitor serial Arduino. Los monitores seriales son un "lazo" entre computadoras. Y tu uno. Le permite enviar y recibir mensajes de texto para facilitar la alineación y el control ¡Uno en el teclado! Por ejemplo, puede enviar una orden desde su ordenador Abre el LED. En este curso, se usará con Lección 15, segunda parte. Por lo tanto, si no lo ha hecho, proceda ahora con la lección 15 de la parte 2.

## Medidas adoptadas

- **Después** de que han subido este cableado sobre el UNO, haga clic en el botón derecho en la barra de herramientas en el IDE de Arduino. Es en un círculo el botón a continuación.
- **Se** abrirá la siguiente ventana. Haga clic en el botón Serial Monitor para encender el monitor serie. Se introducen los conceptos básicos sobre el monitor serial en detalles en la lección 4 parte 2.
- **Esta** ventana se llama al Monitor Serial y es parte del software del IDE de Arduino. Su trabajo es la que permite a ambos enviar mensajes desde tu ordenador a una placa UNO (por USB) y también recibir mensajes de la placa UNO.
- **El** mensaje "Enter LED número 0 a 7 or 'x' para borrar" ha sido enviado por el Arduino. Nos está diciendo qué comandos podemos enviarnos a Arduino: o enviar la 'x' (para apagar todos los LEDs) o el número de LED que desea activar (donde 0 es el LED de la parte inferior, 1 es la siguiente, hasta 7 para el LED superior).
- **Trate** de escribir los siguientes comandos en la parte superior del Monitor Serial que es el nivel con el botón 'Enviar'. Presione 'Enviar', después de escribir cada uno de estos caracteres: x 0 3 5
- **Escribir** x voluntad no tienen ningún efecto si los LEDs ya están todos fuera, pero al entrar en cada número, el correspondiente LED deberá encenderse y usted recibirá un mensaje de confirmación de la placa UNO. El Monitor Serial aparecerá como semuestra a continuación.
- **Escriba** x otra vez y pulse 'Enviar' para apagar todos los LEDs



## Código

- Después del montaje, abra el programa en la carpeta "The\_Serial\_Monitor" donde se encuentra el curso y haga clic en CARGAR para cargar el programa. Consulte la Lección 5 en la parte 1 para obtener detalles sobre la carga del programa si hay algún error.
- Como es de esperar, el boceto se basa en el que se utilizó en la lección 22 de la parte 2. Así que lo haremos. Aquí está la nueva sección. Encontrará útil la referencia a un boceto completo en Arduino IDE.
- En la función de 'configuración', hay tres nuevas líneas al final:
- En primer lugar, tenemos el comando 'Serial.begin(9600)'. Esto inicia la comunicación serial, para que la UNO puede enviar comandos a través de la conexión USB. El valor 9600 es la configuración velocidad de la conexión. Esto es la rapidez con la que los datos debe ser enviado. Esto puede cambiar a un valor más alto, pero también tendrás que cambiar al monitor de Arduino Serial el mismo valor. Hablaremos de esto más adelante; por ahora, dejar en 9600.
- El comienzo de la línea con 'mientras' asegura que hay algo en el otro extremo de la conexión USB para Arduino hablar antes de que comience el envío de mensajes. De lo contrario, el mensaje puede ser enviado, pero no aparece. Esta línea es realmente sólo es necesaria si está utilizando a un Arduino Leonardo porque el Arduino UNO se restablece automáticamente la placa Arduino al abrir el Monitor de la serie, mientras que esto no sucede con el Leonardo.
- La última de las nuevas líneas en 'configuración' envía el mensaje que vemos en la parte superior del Monitor serie.
- The 'loop' function is where all the action happens:
- Todo lo que ocurre dentro del bucle está contenido dentro de una instrucción 'if'. Así que a menos que la llamada a la función incorporada de Arduino 'Serial.available()' es 'true' entonces nada sucederá.
- Serial.Available() devuelve 'true' si los datos ha sido enviado a la ONU y allí está listos para ser procesado. Los mensajes entrantes se llevan a cabo en lo que se llama un búfer y Serial.available() devuelve true si ese buffer es no vacía.

```
void setup()
{
  pinMode(latchPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
  updateShiftRegister();
  Serial.begin(9600);

  while (! Serial); // Wait until Serial is ready - Leonardo
  Serial.println("Enter LED Number 0 to 7 or 'x' to clear");
}
```

```
void loop()
{
  if (Serial.available())
  {
    char ch = Serial.read();
    if (ch >= '0' && ch <= '7')
    {
      int led = ch - '0'; bitSet(leds, led);
      updateShiftRegister();
      Serial.print("Turned on LED ");
      Serial.println(led);
    }
    if (ch == 'x')
    {
      leds = 0; updateShiftRegister();

      Serial.println("Cleared");
    }
  }
}
```

- Si un mensaje ha sido recibido, es a la siguiente línea decódigo:
- Esto lee el siguiente carácter del búfer y elimina del buffer. También asigna a la variable 'ch'. La variable 'ch' es de tipo 'char' que significa 'carácter' y como su nombre indica, tiene un carácterúnico.

```
char ch = Serial.read();
```
- Si usted ha seguido las instrucciones en el prompt en la parte superior del Monitor Serial, luego este personaje serán o bien un número dígito entre 0 y 7 o la letra 'x'.
- La instrucción 'if' en la línea siguiente comprueba para ver si es un solo dígito por ver si 'ch' es mayor o igual que el carácter '0' y menor o igual que el personaje 7'. Parece un poco extraño comparar caracteres de esta manera, pero es perfectamente aceptable.
- Cada carácter está representado por un número único, conocido su valor ASCII. Esto significa que cuando se comparan caracteres usando `<=` y `>=` es realmente los valores ASCII que se estabancomparando.
- Si pasa la prueba, llegamos a la siguiente línea:

```
int led = ch - '0';
```
- ¡Ahora estamos actuando aritmética en los personajes! Estamos restando el dígito '0' de cualquier dígitos fue introducido. Por lo tanto, si escribió '0' y luego '0' a '0' será igual a 0. Si escribió '7' y '7'-'0' será igual al número 7 ya que es realmente los valores ASCII que se utilizan en lasustracción.
- Desde sabemos que el número del LED que queremos encender, nos basta establecer este bit en la variable 'leds' y actualizar el registro de desplazamiento.

```
bitSet(leds, led);  
updateShiftRegister();
```
- Las dos líneas escriben de nuevo un mensaje de confirmación en el Monitor serie.
- La primera línea utiliza `Serial.print` en lugar de `Serial.println`. La diferencia entre los dos es que `Serial.print` no se inicia una nueva línea después de imprimir lo que está en su parámetro. Usamos esto en la primera línea, porque estamos impresión el mensaje en dos partes. En primer lugar el general bits: 'Enciende LED' y luego el número del LED.

```
Serial.print("Turned on LED ");  
Serial.println(led);
```
- El número del LED se realiza en un 'int' variable en lugar de ser una cadena de texto. `Serial.Print` puede tomar ya sea una cadena de texto dentro de comillas dobles, o un 'int' o para el caso casi cualquier tipo de variable.
- Después de la instrucción 'if' que maneja el caso, cuando un dígito se ha manejado, hay una segunda instrucción 'if' que comprueba si la 'ch' es la letra 'x'.

```
if (ch == 'x')  
{  
  leds = 0;  
  updateShiftRegister();  
  Serial.println("Cleared");  
}
```
- Si es así, entonces se borran todos los LEDs y envía un mensaje de confirmación.