

## La función `shiftOut ()`

### Descripción

Desplaza hacia la salida un byte de datos un bit a cada vez. Comienza a partir del bit más significativo (más a la izquierda) o menos significativo (más a la derecha). Cada bit se escribe a su vez en un pin de datos (En nuestro caso el SER), después de lo cual un pulso de reloj se activa (pasa a nivel alto, y a continuación, a nivel bajo) para indicar que el bit está disponible (En nuestro programa ese reloj se representa como SRCLK). Por lo cual debemos asegurarnos que el pin de reloj está en bajo antes de ejecutar la función `shiftOut ()` con una llamada a la función `digitalWrite (clockPin, LOW)`, en nuestro caso (`digitalWrite (SRCLK, LOW)`)

### Sintaxis:

```
shiftOut (dataPin, clockPin, bitOrder, valor)
```

### Parámetros:

*dataPin*: el pin que será usado para emitir cada bit (En nuestro caso el pin [SER](#) de nuestros 74HC595). Tipo: *int*

*clockPin*: el pin usado para enviar la señal de conmutación (flanco de subida) una vez que el *dataPin* ([SER](#)) está emitiendo el bit correcto (En nuestro caso el pin [SRCLK](#)). Tipo: *int*

*bitOrder*: el orden en que los bits serán emitidos; sea MSBFIRST (empezando por el de mayor peso y siguiendo hacia la derecha) o LSBFIRST (comenzando por el de menor peso y siguiendo hacia la izquierda). En nuestro caso usaremos el MSBFIRST por la organización de los leds a controlar y así cada representación de bits corresponda a un led en su respectivo orden.

*valor*: el octeto que será emitido en cada fila para cambiar el estado de los leds previamente organizados (En nuestro programa valor es llamado SFn con  $0 < n < 9$ .) Tipo: *byte*

### Retorna:

Nada

## Notas:

Los pins *dataPin*([SER](#)) y *clockPin* ([SRCLK](#)) deben de ser configurados como salidas mediante una llamada a la función *pinMode* () antes de ejecutar *shiftOut* ().

La función *shiftOut* puede emitir solamente un byte (octeto = 8 bits) cada vez que es ejecutada, por lo tanto, el máximo valor que puede ser ingresado en cada circuito integrado 74HC595 es 11111111 (*representación binaria*) o 255 (*representación decimal*) que en nuestro programa se representaría como un estado alto en todos los leds de la fila que controla ese integrado.

### Implementación de la función *shiftOut* () en nuestro programa

```
//Funcion para controlar los datos de los 8 CI
void ledLineWrite(int Sf1, int Sf2, int Sf3, int Sf4, int Sf5, int Sf6, int Sf7, int Sf8)
{
    shiftOut(SER, SRCLK, MSBFIRST, Sf8);
    shiftOut(SER, SRCLK, MSBFIRST, Sf7);
    shiftOut(SER, SRCLK, MSBFIRST, Sf6);
    shiftOut(SER, SRCLK, MSBFIRST, Sf5);
    shiftOut(SER, SRCLK, MSBFIRST, Sf4);
    shiftOut(SER, SRCLK, MSBFIRST, Sf3);
    shiftOut(SER, SRCLK, MSBFIRST, Sf2);
    shiftOut(SER, SRCLK, MSBFIRST, Sf1);
    digitalWrite(RCLK, HIGH);
    digitalWrite(RCLK, LOW);
}
```

Cada vez que se ejecuta la función *shiftOut* () el octeto de bits es enviado a la salida invertida del CI y de allí se envían al SER del CI siguiente (Tenemos 8 CI 74HC595 conectados en serie)