

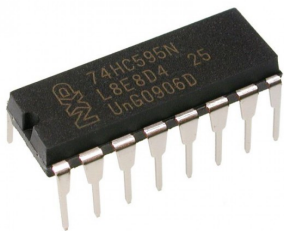
8. Electrónica Digital y el Shift Register

Tallerine Arduino/LED

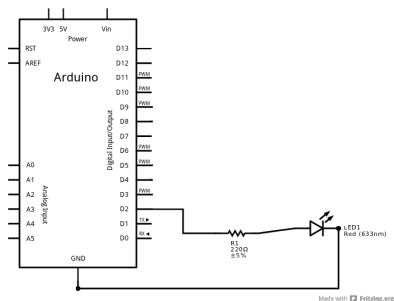
Instituto de Ingeniería Eléctrica

8 de mayo de 2017

Segunda parte del curso



shift registers, electrónica digital



- Arduino nos limita a un máximo de 14 pines (LEDs)
- Pero podemos querer contrlar **más**
- Bicicleta: por lo menos **dos** hileras de LEDs!

Shift register (SR)

- Circuito integrado digital
- **register**: almacena N bits
- **shift**: con una señal, desplaza valores un lugar
- **74HC595**: shift register de 8 bits
- Se pueden **encadenar** para almacenar más bits!
- Sólo requiere **tres** pines del Arduino ...
- Pero es más difícil de programar que manejar los pines directamente
- Hoy veremos cómo

Idea

- Vamos a conectar el Arduino al SR
- Reprogramar Arduino para mostrar hilera de LEDs a través del SR

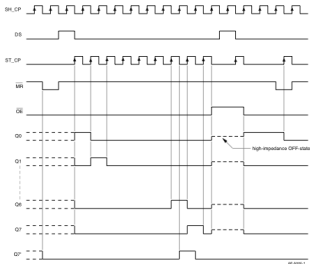
Pasos a seguir

- 1 Aprender algunos conceptos de **electrónica digital**
- 2 Estudiar funcionamiento del **74HC595**
- 3 Aprender cómo **manejar al SR**
- 4 Escribir una función que muestre una hilera de LEDs
- 5 Probarlo! Vamos a rehacer el entregable 1 con un SR

Todo esto está sacado de:

<https://www.arduino.cc/en/tutorial/ShiftOut>

Electrónica digital



FUNCTION TABLE

See note 1.

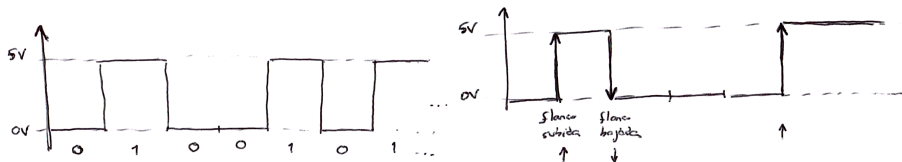
		INPUT				OUTPUT		FUNCTION
SH_CP	ST_CP	OE	MR	DS	Q7	Qn		
X	X	L	L	X	L	n.c.	a LOW level on MR only affects the shift registers	
X	↑	L	L	X	L	L	empty shift register loaded into storage register	
X	X	H	L	X	L	Z	shift register clear; parallel outputs in high-impedance OFF-state	
↑	X	L	H	H	Q6'	n.c.	logic high level shifted into shift register stage 0; contents of all shift register stages shifted through, e.g. previous state of stage 6 (internal OE) appears on the serial output (Q7)	
X	↑	L	H	X	n.c.	Qn'	contents of shift register stages (internal Qn) are transferred to the storage register and parallel output stages	
↑	↑	L	H	X	Q6'	Qn'	contents of shift register shifted through; previous contents of the shift register is transferred to the storage register and the parallel output stages	

Note

1. H = HIGH voltage level;
L = LOW voltage level;
↑ = LOW-to-HIGH transition;
↓ = HIGH-to-LOW transition;
Z = high-impedance OFF-state;
n.c. = no change;
X = don't care.

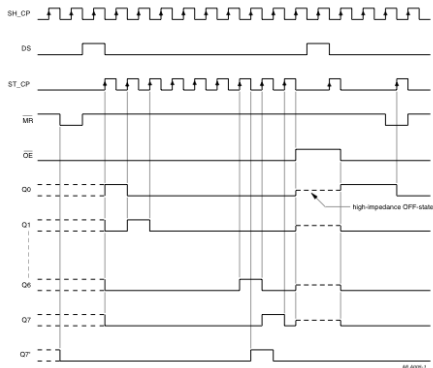
- Los voltajes en un circuito digital se interpretan como 0 o 1
- Usualmente 0 se asocia con 0V o GND (tierra)
- Usualmente 1 se asocia con 5V (TTL – Arduino) o 3,3V (CMOS)
- Pero puede ser otra cosa! (ej. RS-232 usa $-12V$ y $12V$)
- Lo que importa es que hay sólo dos niveles lógicos, 0 y 1
- Vimos que en la práctica se fija un *umbral*, usualmente la mitad del valor de 1
 - en el Arduino, 2,5V

Dinámica de circuitos digitales: niveles y umbrales

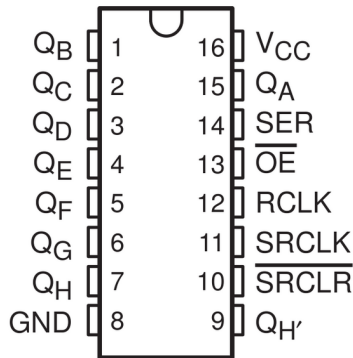


- 0 y 1 son llamados *niveles lógicos*
- Circuitos digitales: pueden depender de valores fijos en un cierto nivel
- Pero pueden depender también de **cambios** en los niveles
- Llamamos a esos cambios **flancos**
- **flanco de subida**: pin o línea pasa de 0 a 1
- **flanco de bajada**: pin o línea pasa de 1 a 0

- Usualmente los flancos se asocian con *eventos*
- El evento más común es un *tick de reloj*
- Los circuitos digitales funcionan al compás de un *reloj maestro*
- Reloj maestro: pulsos a intervalos regulares
- Determina velocidad del sistema
- Arduino: reloj de 48MHz
- Puede haber *otros* relojes



- Los pines de un chip se identifican con un nombre
- Si contiene *ck* o *clk*, es un reloj
- Si tiene una barrita arriba de su nombre, quiere decir que está invertido, es decir, que está *prendido* cuando su valor es 0
- (lo “normal” es que un pin esté prendido cuando vale 1)



FUNCTION TABLE

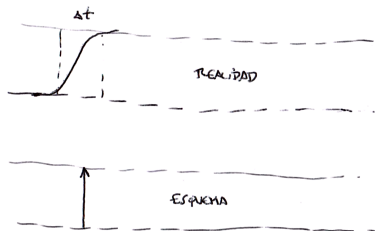
See note 1.

INPUT					OUTPUT		FUNCTION
SH_CP	ST_CP	OE	MR	DS	Q7'	Qn'	
X	X	L	L	X	L	n.c.	a LOW level on MR only affects the shift registers
X	↑	L	L	X	L	L	empty shift register loaded into storage register
X	X	H	L	X	L	Z	shift register clear; parallel outputs in high-impedance OFF-state
↑	X	L	H	H	Q6'	n.c.	logic high level shifted into shift register stage 0; contents of all shift register stages shifted through, e.g. previous state of stage 6 (internal Q6') appears on the serial output (Q7')
X	↑	L	H	X	n.c.	Qn'	contents of shift register stages (internal Qn') are transferred to the storage register and parallel output stages
↑	↑	L	H	X	Q6'	Qn'	contents of shift register shifted through; previous contents of the shift register is transferred to the storage register and the parallel output stages

Note

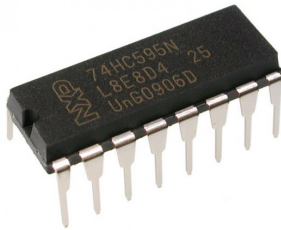
- H = HIGH voltage level;
 L = LOW voltage level;
 ↑ = LOW-to-HIGH transition;
 ↓ = HIGH-to-LOW transition;
 Z = high-impedance OFF-state;
 n.c. = no change;
 X = don't care.

Concepto clave: el retardo

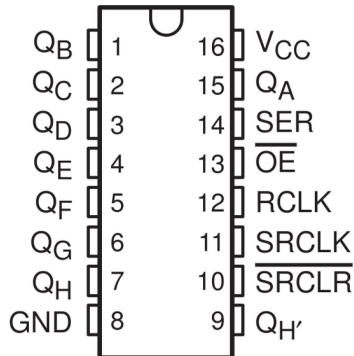


- Nada es realmente instantáneo!
- Los pines demoran en cambiar de estado
- Los flancos toman un cierto tiempo
- Estos tiempos se propagan dentro del circuito
- Y hay que tenerlos en cuenta!
- En nuestro caso es simple, pero hay que estar atentos!

El 74HC595



- QA a QH: 8 salidas del SR
- QH': igual que QH, se usa para enganchar con otro 74HC595
- SER: dato (bit) de entrada
- \overline{OE} (Output Enable) habilita salidas
- RCLK (Register Clock): copia registros del SR a las salidas
- SRCLK (Shift Register Clock): desplaza registros del SR un lugar, entra valor de SER a primer registro
- \overline{SRCLR} (SR Clear): pone registros a 0



FUNCTION TABLE

See note 1.

INPUT					OUTPUT		FUNCTION
SH_CP	ST_CP	OE	MR	DS	Q7'	Qn'	
X	X	L	L	X	L	n.c.	a LOW level on MR only affects the shift registers
X	↑	L	L	X	L	L	empty shift register loaded into storage register
X	X	H	L	X	L	Z	shift register clear; parallel outputs in high-impedance OFF-state
↑	X	L	H	H	Q6'	n.c.	logic high level shifted into shift register stage 0; contents of all shift register stages shifted through, e.g. previous state of stage 6 (internal Q6') appears on the serial output (Q7')
X	↑	L	H	X	n.c.	Qn'	contents of shift register stages (internal Qn') are transferred to the storage register and parallel output stages
↑	↑	L	H	X	Q6'	Qn'	contents of shift register shifted through; previous contents of the shift register is transferred to the storage register and the parallel output stages

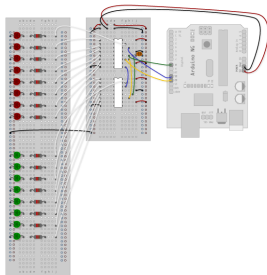
Note

- H = HIGH voltage level;
 L = LOW voltage level;
 ↑ = LOW-to-HIGH transition;
 ↓ = HIGH-to-LOW transition;
 Z = high-impedance OFF-state;
 n.c. = no change;
 X = don't care.

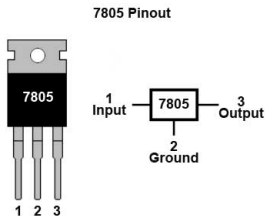
Ejercicio en clase

Conectar arduino con el SR de la siguiente manera:

- VCC de Arduino a VCC de SR
- GND de Arduino a GND de SR
- Pin 2 a SRCLR
- Pin 3 a SER
- Pin 4 a SRCLK
- Pin 5 a RCLK
- GND a OE
- Conectar 8 LEDs (con sus resistencias) a QA, QB, ... , QH
- Bajar, inspeccionar y cargar el programa [sr0.ino](#) desde el EVA
- Ejecutarlo y entender cómo funciona



- Se puede usar el último bit (QH') como bit de entrada a otro SR!
- Así se pueden conectar todos los que se quiera
- El límite es la potencia que se pueda manejar,
- y el retardo admisible en el desplazamiento



ATENCIÓN

- Puede que el Arduino no aguante 2+ 74HC595 y 16+ LEDs
- En ese caso, se usa un regulador de voltaje externo para los 5V de todo el circuito!
- Veremos en clase cómo se usa

Entregable

- Repetir el entregable 3 “Animación con LEDs” usando el SR
- Extra: Quién se anima a usar los 8 LEDs?
- Extra: Quién se anima a usar 16 LEDs??