**Manejo de pines eficientes de Arduino**

En Arduino tenemos la limitante de contar con una humilde cantidad de entradas y salidas digitales, así como también de sus entradas analógicas y sus salidas con PWM, si bien es cierto que otros modelos como el Arduino mega basado en una ATMEGA2560 cuenta con 4 veces mas de estos pines de lo que es común en muchos casos usar este Arduino para un fin tan simple como usar una matriz grande no tiene sentido, por ello recurrimos al manejo de diversos integrados para esta labor así dejamos de ser solo programadores y nos metemos mas a fondo en el mundo de la electrónica verdadera para nuestra felicidad y tristeza.

¿Qué opciones tenemos?

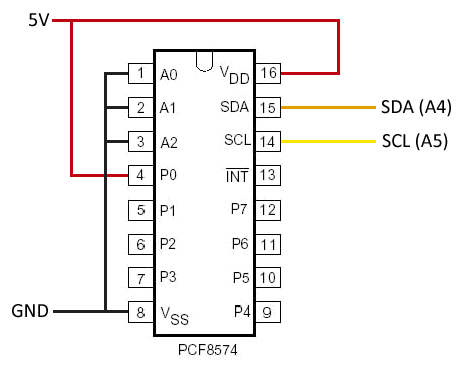
Existe infinidad de modelos con los cuales podemos trabajar para diversas tareas, pero en un nivel general y básico en esta parte hablaremos de los mas usados y simples lo cual ayuda dado que muchos programadores de alto nivel hacen bibliotecas con poderosas herramientas para trabajar con ellos.

Las enlistare en orden de mis favoritas personales:

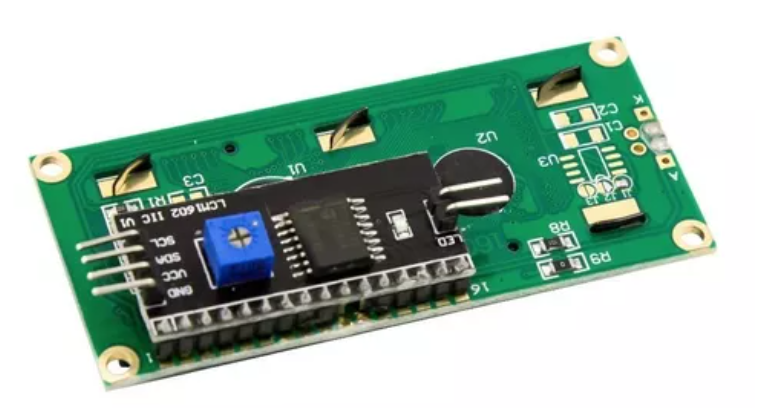
* PCF8574
* 74HC595
* MAX7219
* CD4017

¿Qué nos ofrece cada uno?

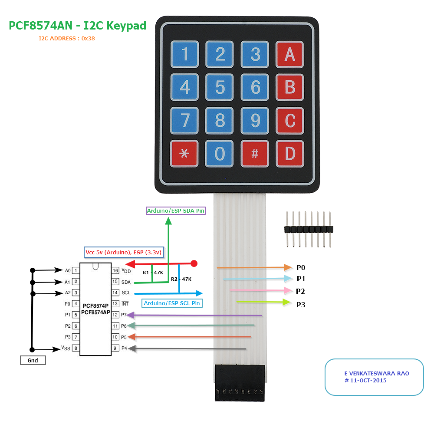
**PCF8574 el ampliador de pines**

Este integrado es de los más útiles de mis favoritos por todo lo que se puede hacer con el y su manejo, existiendo versiones similares, este es el precursor originalmente creado por philips nos permite agrandar la cantidad de salidas y entradas que podemos tener en nuestro Arduino atreves del bus de comunicaciones I2C lo cual lo hace extremadamente versátil dado que solo necesita dos cables de comunicación

Su uso más común y que seguro nunca lo notaron es en el de las pantallas LCD para evitar que tener que malgastar tantos pines del Arduino al pedo

Otro uso también y el mas útil es en el de manejo de matrices de botones como una botonera de teclado



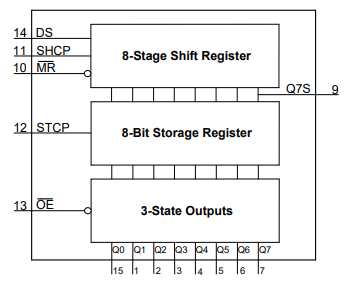
Todo eso se debe a que el PCF se puede configurar por código para que sus patillas funcionen como salida como entrada en vez de solo como salida a diferencia de los otros integrados, es como el terreneitor de los integrados, es por eso que se puede usar en la matriz dado que 4 patillas funcionan como entrada, y las otras 4 como salida

**74HC595 la humildad hecha integrado**

Este integrado de lo mas simple con 3 pines digitales y una lógica muy simple puede manejar cuantas salidas de tu imaginación y hacer una conversión serie paralelo de lo mas elegante, es en palabras redundantes en registro de desplazamiento con un poco de esteroides.

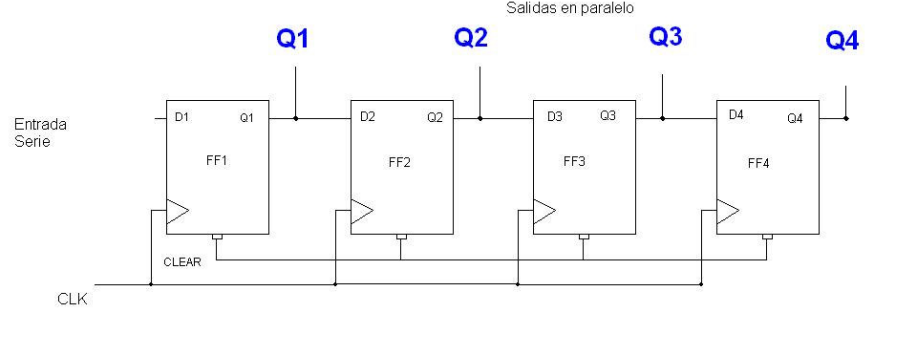
(Es importante meterle onda a este integrado ya que siembra las bases de conocimiento para poder entender como funciona todo este mudillo en general)

Si vemos la hoja de datos veremos este diagrama de funcionamiento, normalmente no nos metemos en la teoría, pero en este caso creo que es importante dado que no es complicada y nos es muy útil.

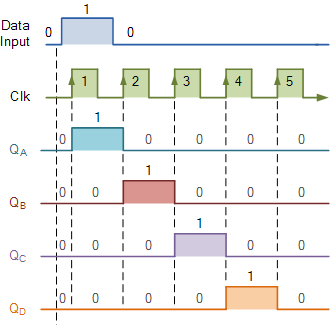
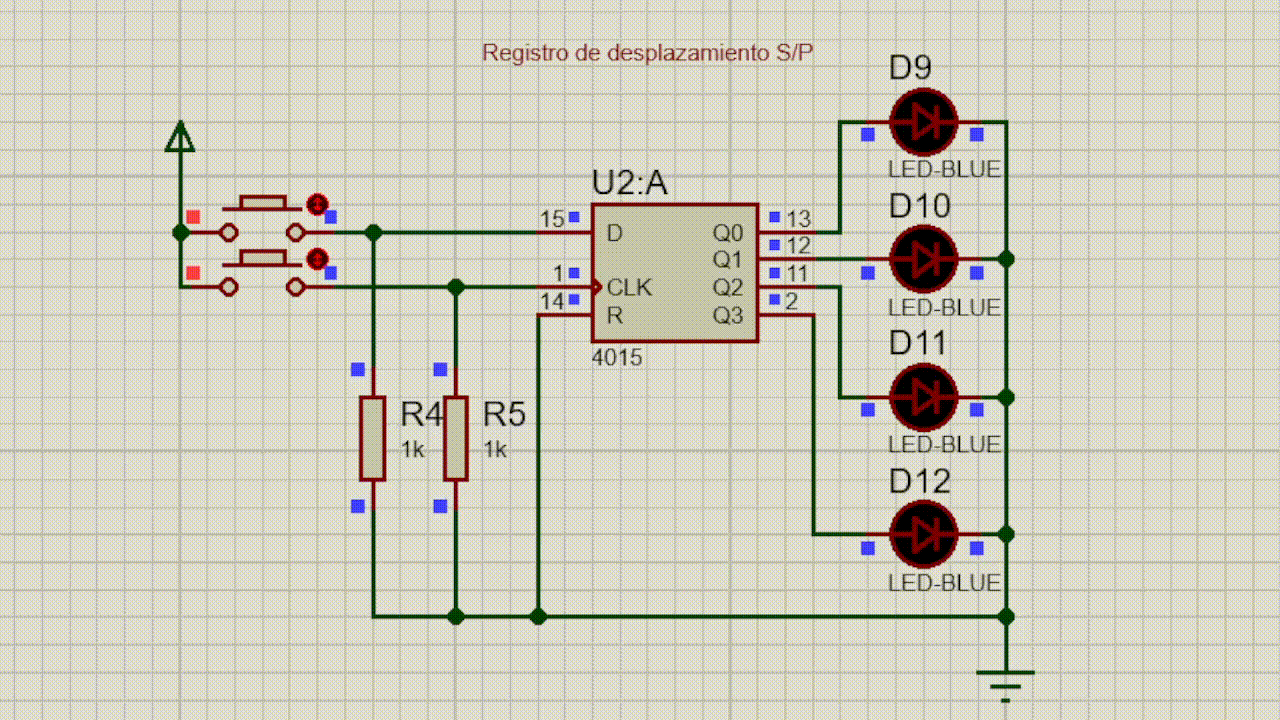


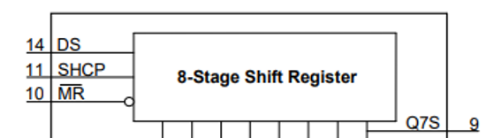
Vayamos por partes, primero el registro de desplazamiento:

Cuando pensamos en un registro de desplazamiento (en este caso serie paralelo) se nos viene a la mente estos pesados diagramas

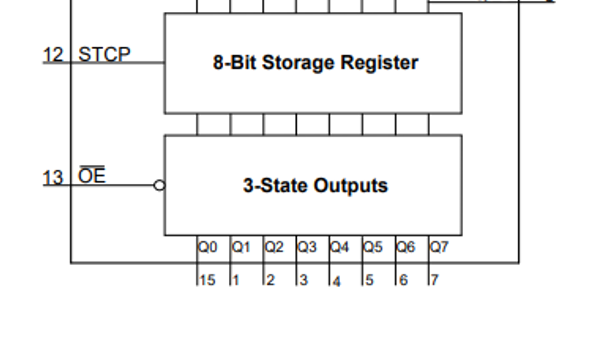


Pero es mas simple de lo que parece, es un circuito que los valores de sus salidas dependen de sus entradas y de los valores anteriores los cuales se manejan de forma sincrónica normalmente, este es el llamado clock el cual marca el paso al que trabaja los datos, es como el tambor que marca cuando un soldado debe marchar, en este caso según los pulsos de este reloj no indicara cuando debe el pulso marcar tomado del dato, por cada clock el pulso sera tomado por la primera salida, y desplazado a la siguiente por el segundo clock y así sucesivamente, si el dato se mantiene positivo tomara el clock lo llevara a cada salida

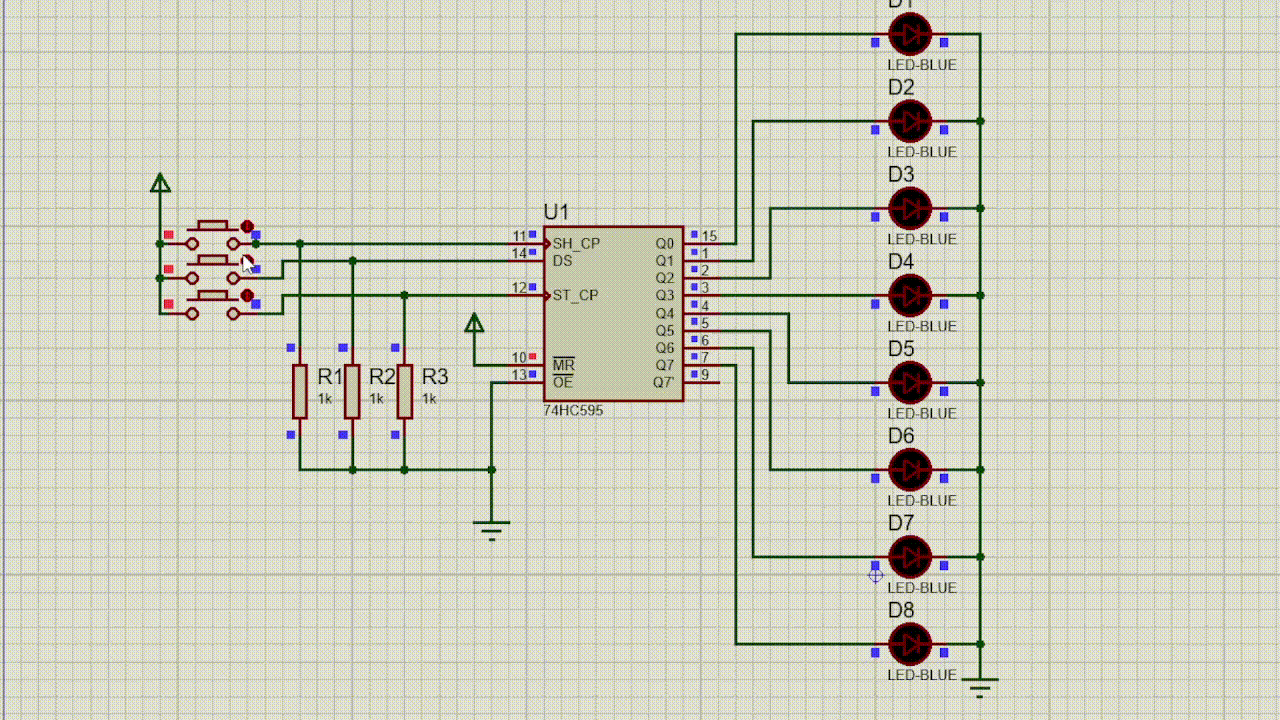
 

En nuestro caso DS son los datos seria que le llega, SHCP es el clock del registro serie y MR nos es el reseteo el cual se activa con un estado bajo por lo tanto buscaremos siempre tenerlo en estado alto. Q7S será el desbordamiento al final de esta y sirve para concadenar varios integrados

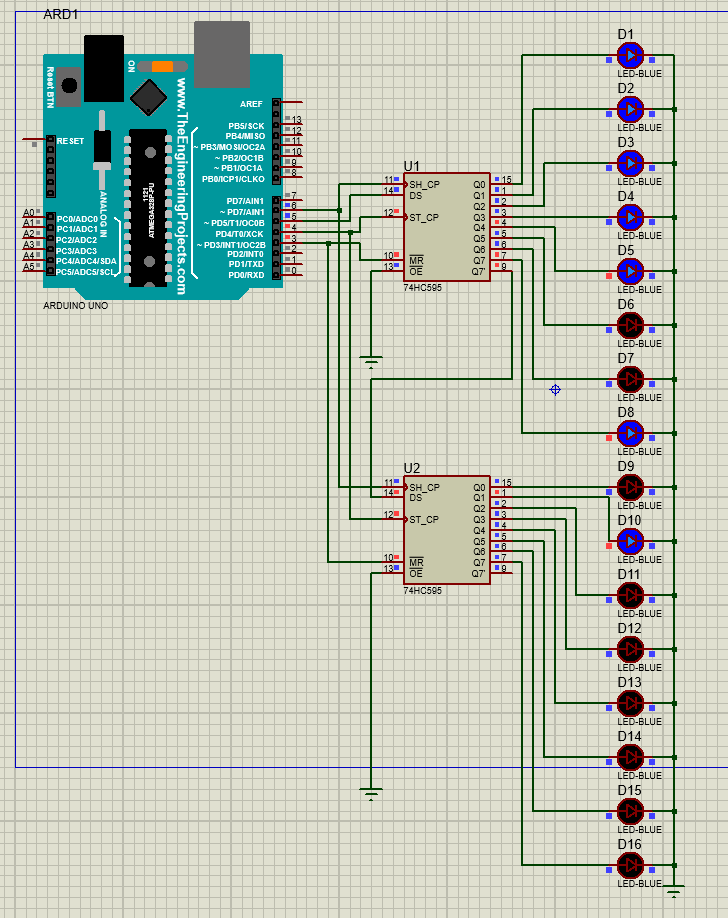
La siguiente parte es mas simple, lo único que hace es guardar el estado en el que se encuentra y se activa al presionar su propio clock el cual es STCP, tomando todos los valores y guardándolos hasta recibir otro pulso, esto se hace para que ningún cambio hecho en el registro de desplazamiento nos afecte a la salida a menos que queramos, y OE lo que hace es activar la salida, si esta está en estado alto no permitirá que haya salida alguna.



Resumiendo, podríamos decir que su lógica de manejo es simple: tenemos 3 pines, 1 de dato (DS), 1 de clock (SHCP) y 1 ultimo de latch (STCP) los cuales harán el trabajo de manejar lo que tú quieras, y los demás solo serán de resteo y activación.



Esta simple base nos da la capacidad de convertir esta salida en lo que queramos y es la base de muchos de los circuitos integrados que usamos diariamente, acá un ejemplo con Arduino que convierte valores seriales en salidas digitales de 2 Byte:



Como pueden apreciar en caso de querer ampliar el numero de patillas de las 8 que trae tenemos que usar Q7S la cual ira en el DS del siguiente integrado, y los demás bits se quedan en paralelo

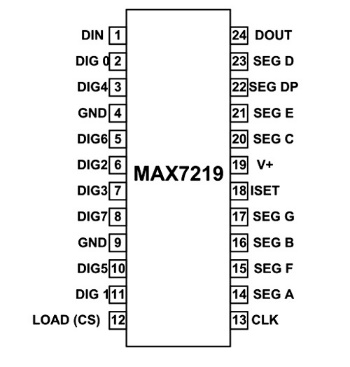
Todos estos ejemplos lo podrán encontrar en GitHub haciendo clic [aquí](https://github.com/Dfcerr/ejemplos-SecArduino-Unitec/tree/main/Manejo%20de%20pines%20eficientes%20de%20Arduino/74HC595) o entrado al apartado de 74HC595

**MAX7219 la evolución**

Este circuito integrado ya es mucho mas avanzado que el 74HC por lo que no nos podremos a ver su teoría a profundidad, este integrado se maneja bajo un protocolo de transferencia SPI aunque en la practica solo usaremos 3 pines y según la biblioteca podremos conectarlo donde queramos sin la necesidad de limitarnos por los pines de SPI del Arduino, este integrado cuenta con dos grandes funciones:

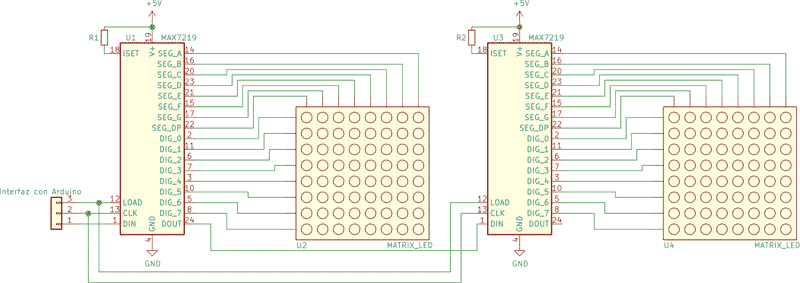
Es capaz de manejar por integrado matrices de 8x8 leds con la posibilidad de concatenar más matrices usando el mismo bus de datos

La segunda es que puede variar digitalmente el brillo de estos leds lo cual es bastante importante, además de tener un modo ahorro según lo necesitemos.

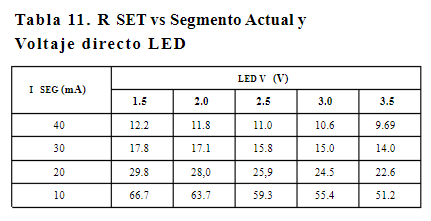
El max cuenta con una serie de patillas, siendo las salidas los pones SEG(A-G), DP y DIG(0-7) (este nombre se debe por que originalmente se lo tiene pensado para manejar display de 7 segmentos).

Luego tenemos a DIN, LOAD y CLK que son los que van conectados al Arduino

También tenemos al pin DIN y DOUT los cuales se pueden conectar para intercalar varias matrices



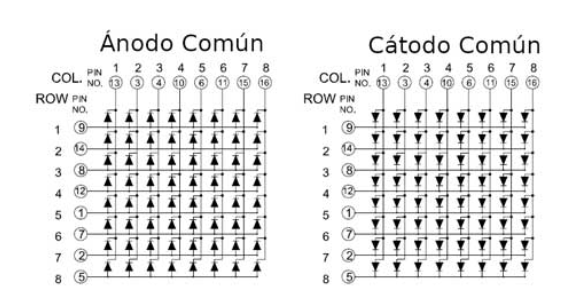
Y por ultimo nos queda el pin mas importante, ISET el cual nos sirve para setear la intensidad máxima a la cual queremos que trabaje nuestros leds, bajo la siguiente tabla:



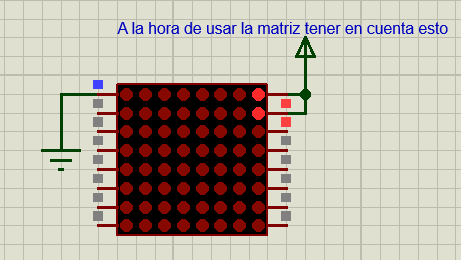
Siendo el valor de las resistencias en KΩ

Esto solo nos dará el máximo de intensidad, pero luego desde código podremos variar este valor desde 0 hasta el máximo.

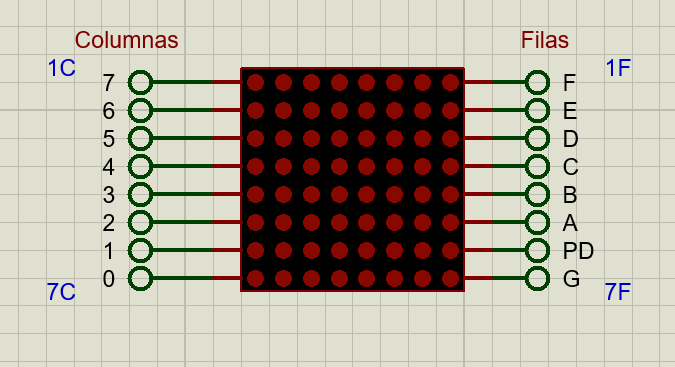
Para colocar nuestra matriz será importante determinar si es de cátodo común o ánodo común, según la que tengamos elegiremos donde irán nuestras filas y columnas.



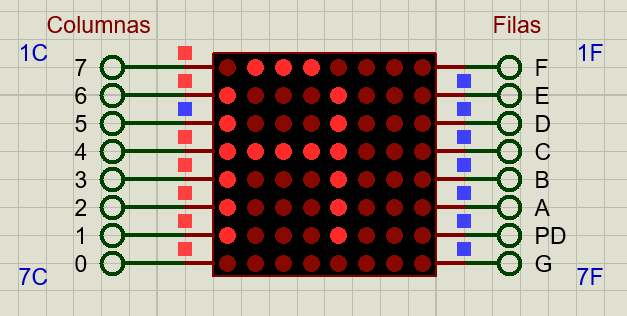
Nosotros elegimos en el momento usar la librería MD\_MAX72XX la cual era bastante potente, pero tenía sus complicaciones. Primero a la hora de realizar las pruebas de orden de conexión tome en cuenta esto:



Si la matriz al conectar dos entradas a positivo y una a negativo buscando que queden los dos leds apuntados arriba a la derecha, bajo esta pauta podremos saber el orden que maneja la biblioteca:

Este será el orden de conexión de las patillas según lo visto antes y que numero tendrá cada uno, siendo la columna 1 y fila 1 la de arriba a la derecha.

Esto se puede cambiar a voluntad, en invirtiendo ordenes de conexión pero si queremos usar la biblioteca de letras que trae la biblioteca tendremos que hacerlo de esta forma.



Aunque ya les digo desde ya que es medio rara, y es preferible hacernos uno mismo está, pero todo dependerá del gusto de cada uno.

Ahora explicare un poco de cómo usarla:

Constructor:

MD\_MAX72XX() es la función encargada de construir el objeto con el cual trabajaremos constantemente y tiene una opción para usar los buses de SPI clásicos de Arduino acá usare otros con la siguiente sobrecarga:

|  |
| --- |
| D\_MAX72XX mx = MD\_MAX72XX (hardware, pinDIN, pinCLK, pinCS, matCount); |

**En hardware directamente colocaremos**: MD\_MAX72XX::DR0CR0RR1\_HW

Los datos de pines serán a donde lo hayamos conectado y matCount será el numero de matriz que tengamos, en este caso estamos usando 1 sola matriz

Setup():

En las funciones de configuración inicializaremos el todo con: mx.begin()

Luego tenemos a mx.control(mat, param, value)

Siendo mat: será el numero de matriz sobre la cual hacemos la configuracion arrancando desde el 0.

En param tenemos que colocar MD\_MAX72XX:: debe ir seguido de lo siguiente haciendo:

|  |  |
| --- | --- |
| ::TEST | Activa o desactiva el modo de prueba. Cuando se activa (value = true) se encienden todos los LEDs para comprobar que ninguno esté dañado. |
| ::SHUTDOWN | Activar o desactivar el modo de bajo consumo |
| ::INTENSITY | Permite especificar el brillo de los leds. El valor de value tiene que estar entre 0-15, donde 0 apaga los leds y 15 es el brillo máximo. |
| ::UPDATE | Permite habilitar o deshabilitar la actualización automática |

Si obviamos colocar mat los cambios se efectúan para todas las matrices.

Luego de eso tenemos las funciones de operación las cuales podemos usarlo donde queremos (las otras también, pero es importante que las declaremos en el setup antes que nada)

* getColumnCount()
* clear()
* setColumn()
* setPoint()
* setRow()
* transform() la cual tiene 8 distintas transformaciones
* setChar()
* update() (solo si desactivamos update automatico)

si quieren saber que hace cada función en específico vayan a la bibliográfica que ahí estará mejor explicado, igual los ejemplos para usarla estarán en la carpeta de este módulo.

Dentro del ejemplo deje 6 ejemplos de usos que se le pueden dar:

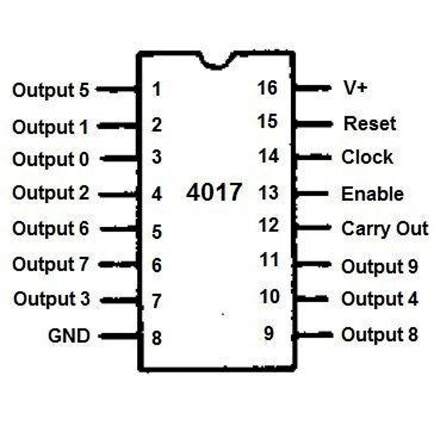
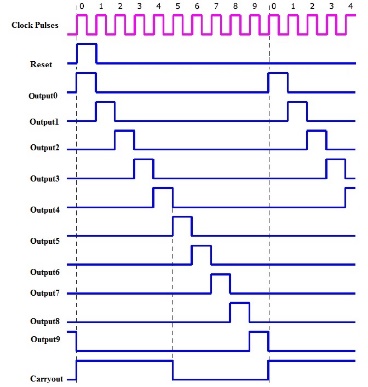
Bibliográfica: [biblioteca](https://github.com/MajicDesigns/MD_MAX72XX/tree/main)

[Fuente](https://hetpro-store.com/TUTORIALES/matriz-led-max7219/#:~:text=El%20MAX7219%20es%20muy%20%C3%BAtil,tienda%20virtual%2C%20puedes%20adquirlo%20aqui.)

GitHub de este aparte clic [aquí](https://github.com/Dfcerr/ejemplos-SecArduino-Unitec/tree/main/Manejo%20de%20pines%20eficientes%20de%20Arduino/MAX7219) o entrando a max7219

**CD4017 secuenciador básico**

Este integrado es muy usado en aplicaciones pequeñas por su simpleza, son de esas cosas básicas de la electrónica que nos dan el pilar para otras aún mayores, también llamado contador de décadas este intrigado al igual que otros contadores a medida que le van llegando los pulsos en su clock este desplaza el valor de salida positivo entre sus salidas

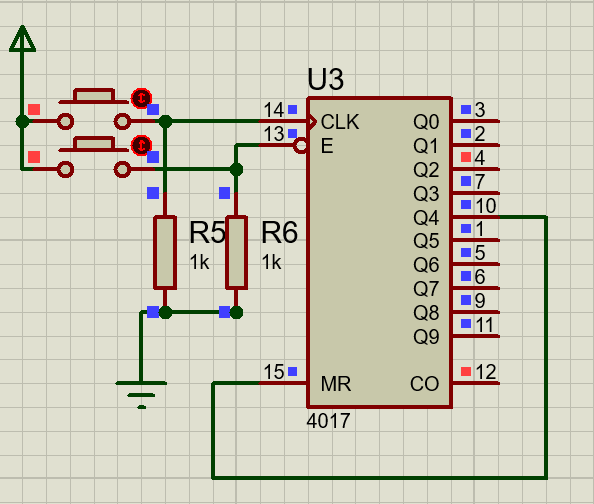
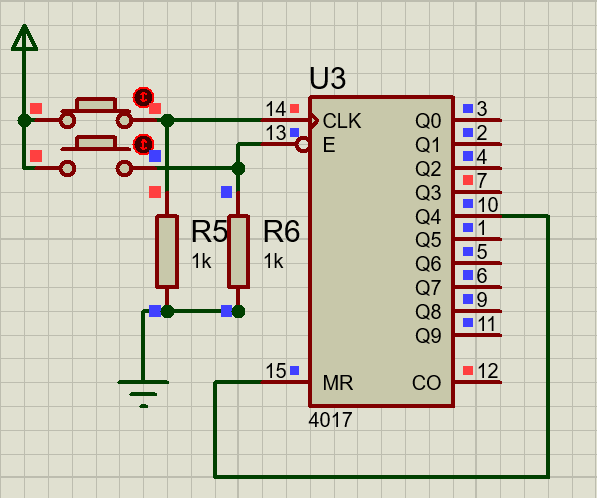
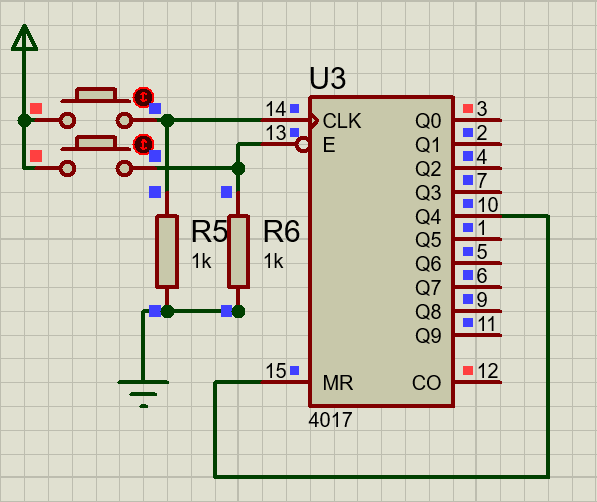
 

Tiene 16 pines, 2 de alimentación, 10 de salidas del contador, el carry out, el enable y el reset

Como ya explicamos a medida que le van llegando los pulsos como se puede ver en el registro de tiempo, para esto el enable debe estar conectado a tierra

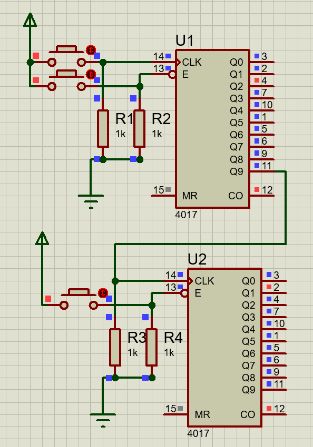
El carry out cambia de estado según el ciclo de 5 en el que este, siendo en los primero 5 que estaré en un estado alto, y después hasta que termine su periodo de 10 estará en estado bajo, por ultimo el reset si se le suministra un estado alto reiniciara al inicio al integrado. Es un circuito de lo más primitivo, pero puede llegar a ser de utilidad con el suficiente ingenio.

Caso 1 conteo de ciclos más cortos:

Como se puede apreciar según donde lo conectes el reset podes obtener un contador de menos paso lo cual para algunos casos nos puede ser de utilidad

Caso 2 contador de decenas

en este caso si concadenas Q9 con la entrada del clock del siguiente integrado conseguís una cascada que te contara 100 veces, si colocas otra obtendrás una de mil

Si mezclas esto con el de caso 1 podes obtener un contador del tamaño que necesites para lo que necesites.

En algunos casos si necesitas que mediante pulsos pase de un estado a otro esto te será lo mas efectivo que tendrás, todo dependerá de lo que necesites hacer y que tan bien lo puedas mezclar con otros.

GitHub de este aparte clic [aqui](https://github.com/Dfcerr/ejemplos-SecArduino-Unitec/tree/main/Manejo%20de%20pines%20eficientes%20de%20Arduino/CD4017) o entrando a CD4017

**Como se dijo anteriormente existe una infinidad de integrados que pueden adaptarse a nuestra necesidad, pero muchos vienen de estas líneas de trabajo a sí que sabiendo usar estas estarán más listo para manejar otras.**