MATRIZ DE LEDES

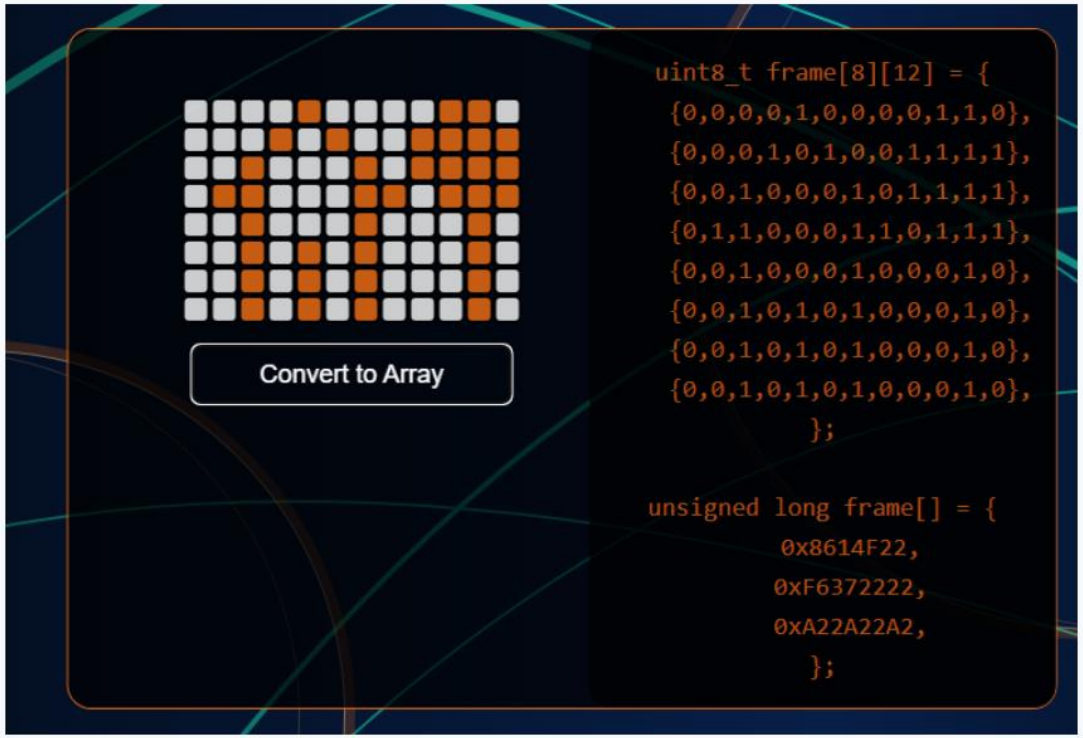
El Arduino UNO R4 WiFi viene con una matriz LED integrada de 12x8 que se puede programar para mostrar una variedad de gráficos, animaciones, etc.

Necesitarás la biblioteca Arduino_LED_Matrix

La bilbioteca ofrece dos formas de controla la matriz:

1. Utiliza una **matriz bidimensional**, con ceros y unos para representar si el LED correspondiente está apagado o encendido. Cada número corresponde a un LED en la matriz LED. Así está hecho el programa **matriz_basica.ino**
2. utilizar una **matriz de enteros de 32 bits** para mantener el estado de la matriz LED. Este método es más complejo. Cada unsigned long almacena 32 bits. Por lo tanto, para una matriz LED de 12x8, que contiene 96 LEDs, necesitarás al menos tres variables unsigned long. Así están hechos los programas **matriz_coracon.ino** y **matriz_latido.ino**

Ahora **vamos a crear nuestros dibujos** para luego verlos en la matriz de Arduino. Para ellos podemos usar **https://www.manualdomaker.com/matrix/**



MATRIZ DE LEDES

```
#include "Arduino_LED_Matrix.h"  //incluimos la librería de la matriz
ArduinoLEDMatrix Pantalla;       //Creación del objeto que será la matriz. La llamamos "pantalla"

byte SAG [8][12] = {           //creamos una matriz byte bidimensional de 8filas y 12 columnas llamada "SAG"
    {1,1,1,0,0,1,0,0,1,1,1,1},  //0= led apagado
    {1,0,0,0,1,0,1,0,1,0,0,0},  //1= led encendido
    {1,1,1,0,1,1,1,0,1,0,1,1},
    {0,0,1,0,1,0,1,0,1,0,0,1},
    {1,1,1,0,1,0,1,0,1,1,1,1},
    {0,0,0,0,0,0,0,0,0,0,0,0},
    {0,0,0,0,0,0,0,0,0,0,0,0},
    {0,0,0,0,0,0,0,0,0,0,0,0},
};
void setup() {

  Pantalla.begin();                    //Inicializamos la matriz de la placa
  Pantalla.renderBitmap (SAG, 8,12);   //llamamos a la matriz creada para imprimir
}
```

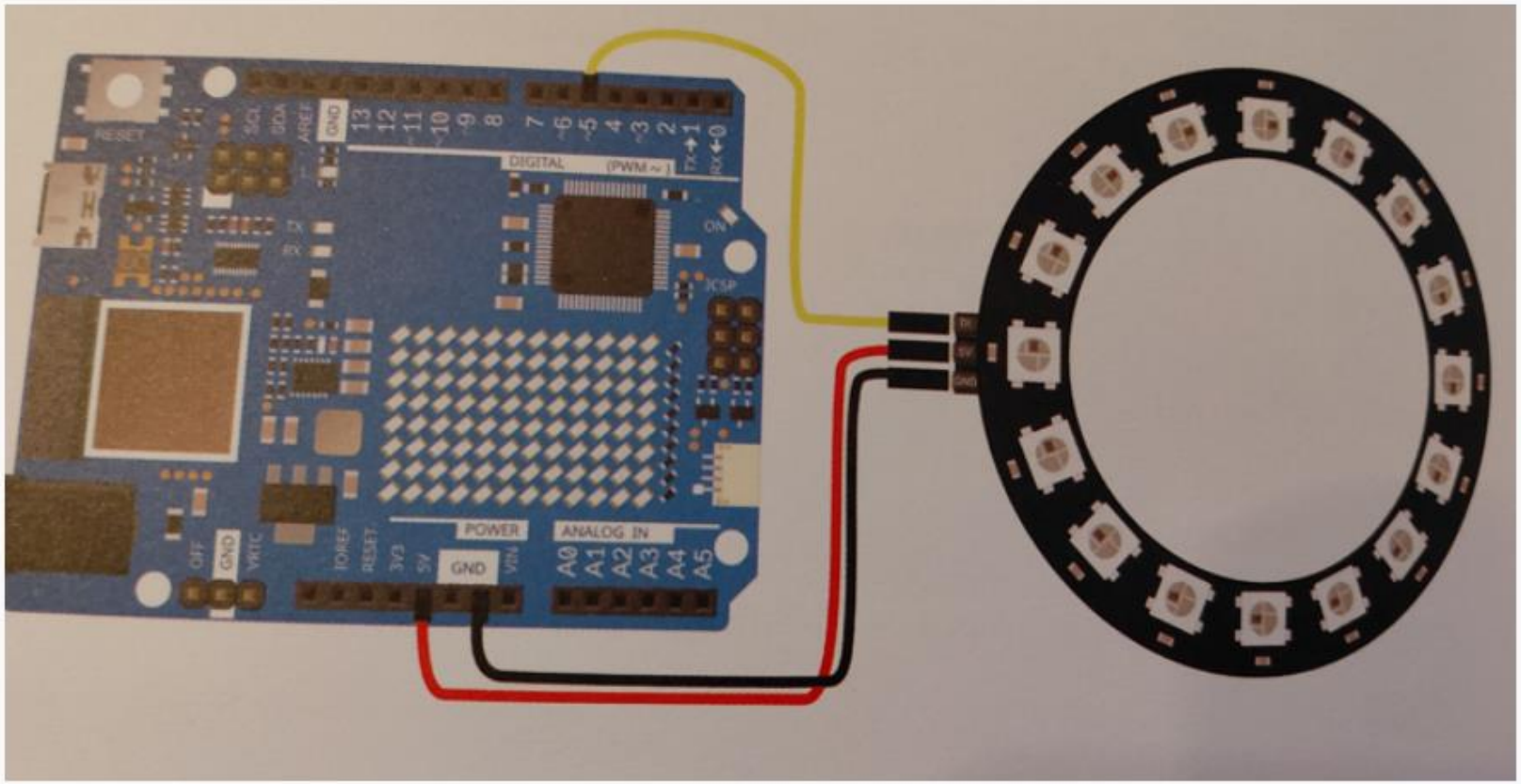
TAREA PROPUESTA: usa la página web anterior para crear una pequeña animación en Arduino (puedes usar el progrma matriz_latido como punto de partida y modificarlo)

ANILLO LED RGB

Un anillo led RGB es un módulo circular compuestro por varios led RGB dispuestos en forma de anillo. Cada led del anillo puede ser controlado de forma independiente para mostrar cualquier color.

Para poder usar el anilo de ledRGB necesitas la librería **Adafruit_neopixel.h**

Conexionado



ANILLO LED RGB

El programa **anilloLED_v1.ino** enciende uno a uno todos los ledes del anillo en color rojo.

TU TAREA: modifica el programa anterior para que los ledes primero se enciendan uno a uno y luego se apaguen uno a uno. Por siempre. Puedes cambiar su color.

```
#include <Adafruit_NeoPixel.h>  //librería para controlar el anillo led RGB

#define ANILLO 5  //pin digital al que está conectado el anillo led RGB

//creamos una instancia del objeto NeoPixel para un anillo de 16 ledes
Adafruit_NeoPixel rgb(16, ANILLO, NEO_GRB + NEO_KHZ800);

int led = 0;              //índice del led
const int espera = 100;  //tiempo en ms de espera para el cambio de color


void setup() {
  rgb.begin();  //inicializa el anillo de ledes
  rgb.setBrightness(60);  //fijamos el brillo del los ledes. Entre 0-255
}

void loop() {
  //vamos a encender los ledes en rojo uno a uno
  for (led=0; led <16; led ++)
  {
    rgb.setPixelColor(led, rgb.Color(255,0,0)); //led rojo
    rgb.show();  //actualiza el anillo para mostrar el color
    delay(espera);
  } //del for
}
```

Aquí te dejo el código RGB de otros colores para que elijas:

- **Rojo (Red):** rgb(255, 0, 0)
- **Verde (Green):** rgb(0, 255, 0)
- **Azul (Blue):** rgb(0, 0, 255)
- **Blanco (White):** rgb(255, 255, 255)
- **Negro (Black):** rgb(0, 0, 0)
- **Amarillo (Yellow):** rgb(255, 255, 0)
- **Cian (Cyan):** rgb(0, 255, 255)
- **Magenta:** rgb(255, 0, 255)
- **Gris (Gray):** rgb(128, 128, 128)
- **Naranja (Orange):** rgb(255, 165, 0)
- **Maroon:** rgb(128, 0, 0)
- **Navy:** rgb(0, 0, 128)