MANUAL

Reloj Arduino nano

By Nicolas Sanabria

ÍNDICE

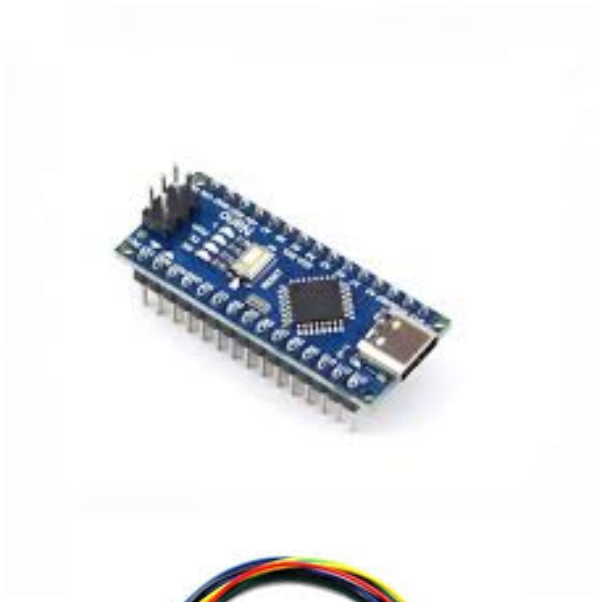
En la antigüedad, medir el tiempo era un desafío que llevó a la creación de dispositivos ingeniosos como el reloj de arena. Este sencillo aparato, compuesto por dos bombillas de vidrio conectadas por un cuello estrecho, permite que la arena fluya de una bombilla a otra en un tiempo específico, generalmente una hora. Su diseño elegante y funcional lo hizo esencial en hogares y navegación.

Con el avance de la tecnología, los métodos para medir el tiempo han evolucionado, pero el encanto del reloj de arena aún fascina a muchos. Este proyecto combina la belleza del reloj de arena tradicional con la precisión de la tecnología moderna, creando un reloj de arena digital con un Arduino Nano.

El Arduino Nano es una pequeña pero poderosa placa de microcontrolador, ideal para proyectos de electrónica debido a su tamaño compacto y capacidad de procesamiento. En este tutorial, utilizaremos el Arduino Nano junto con LEDs, una pantalla LCD o de siete segmentos, resistencias y botones para construir un reloj de arena digital. La programación se realizará con el entorno Arduino IDE.

Te guiaré paso a paso en la selección y conexión de los componentes, así como en la programación del Arduino Nano. Al final, tendrás un reloj de arena digital funcional que combina tradición e innovación tecnológica. Sumérgete en este proyecto y descubre cómo fusionar lo antiguo y lo moderno en una creación única y fascinante

Materiales necesarios para hacer un reloj con Arduino



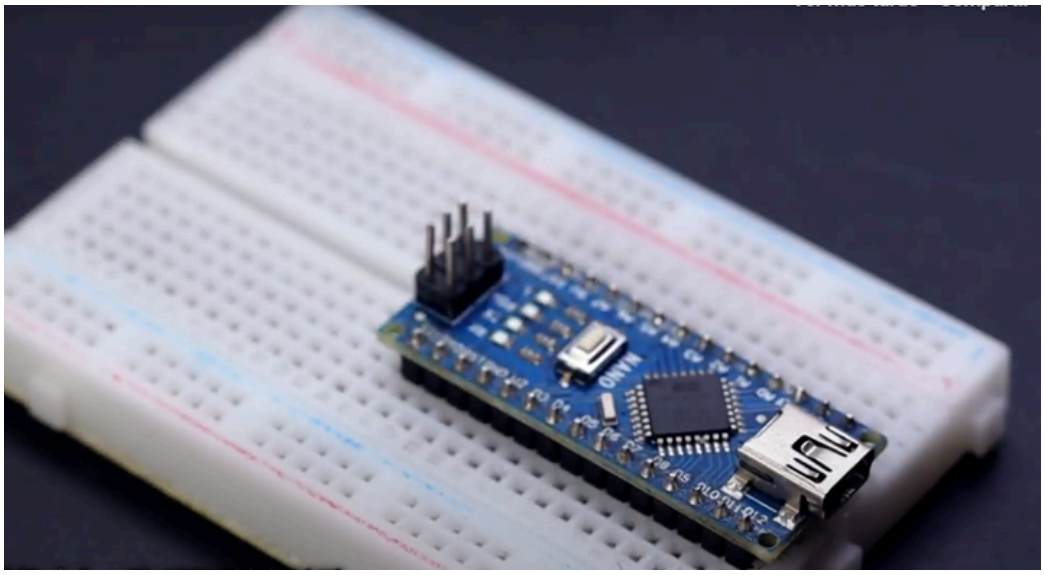
Arduino nano

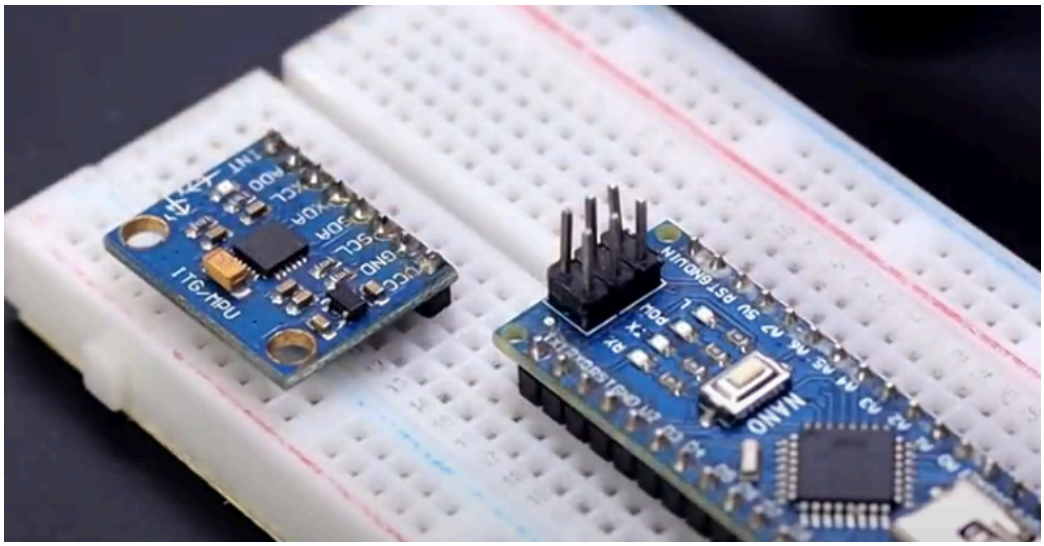
Jumpers

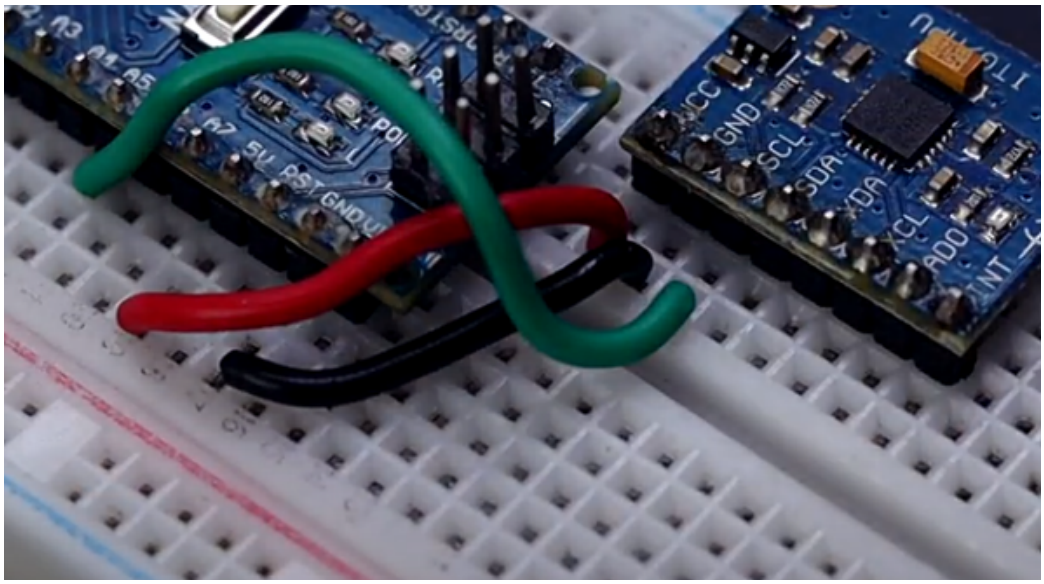
Matrix 8x8

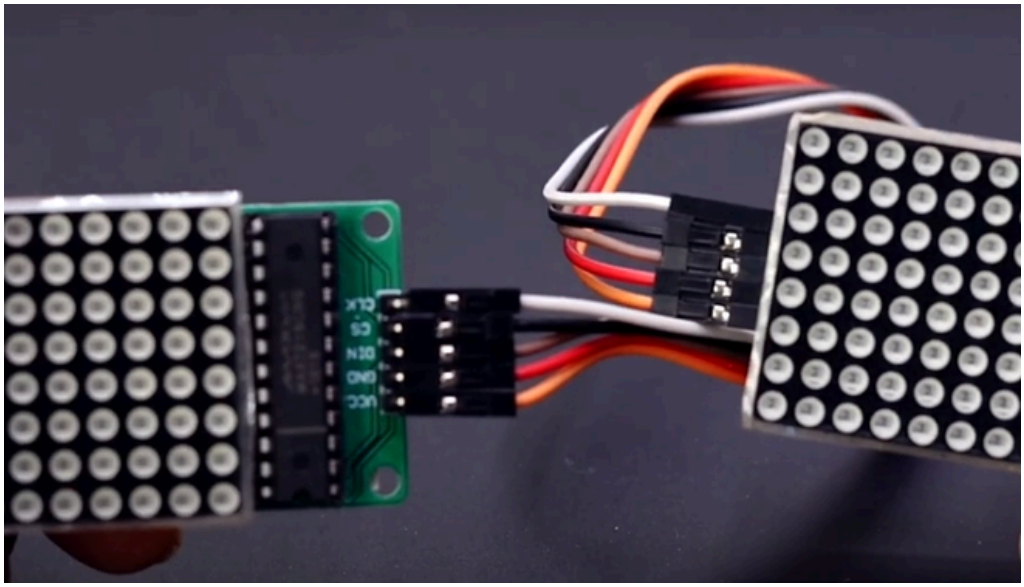
Protoboard

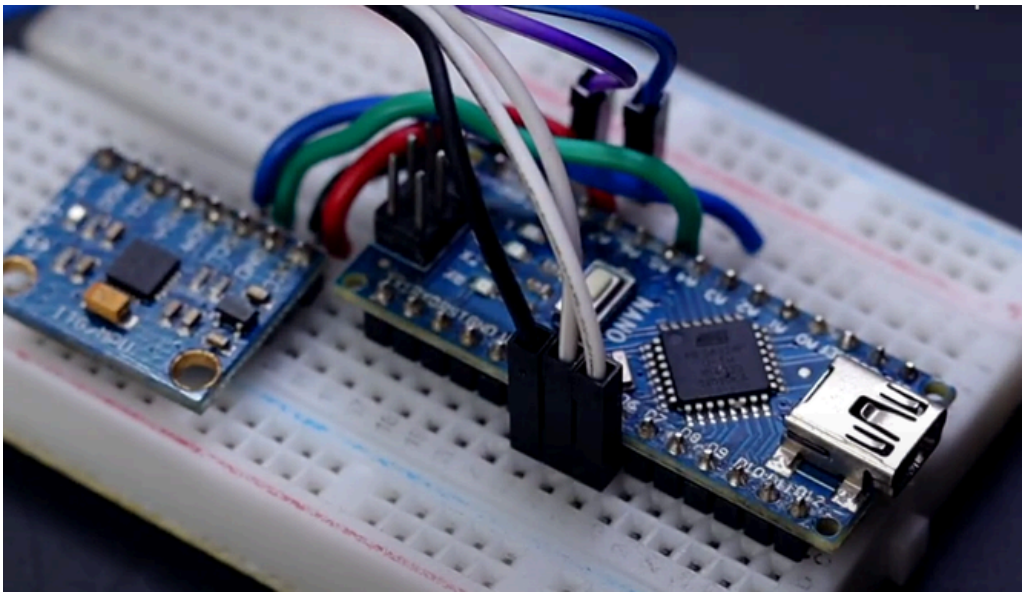
Construcción del circuito








CLK
CS
DIN
GND
VCC



Paso a paso

Materiales Necesarios:

1. Arduino Nano
2. Matriz de LEDs 8x8
3. Jumpers
4. Protoboard
5. Fuente de Alimentación para Arduino Nano
6. Cable USB para Programar el Arduino Nano
7. Software Arduino IDE

Paso 1: Preparación del Espacio de Trabajo

1. Organiza todos los componentes y herramientas en tu área de trabajo para facilitar el acceso.
2. Asegúrate de tener una fuente de alimentación estable y un entorno limpio y ordenado.

Paso 2: Montaje del Arduino Nano en la Protoboard

1. Coloca el Arduino Nano en la protoboard. Asegúrate de que esté firmemente insertado para evitar conexiones sueltas.
2. Conecta los pines GND y 5V del Arduino Nano a las barras de energía de la protoboard para distribuir la energía a otros componentes.

Paso 3: Colocación y Conexión de la Matriz de LEDs 8x8

1. Inserta la matriz de LEDs 8x8 en la protoboard. Asegúrate de que los pines estén correctamente alineados.
2. Conecta el pin VCC de la matriz al pin de 5V del Arduino Nano usando un jumper.
3. Conecta el pin GND de la matriz al pin GND del Arduino Nano.

- Conecta el pin DIN de la matriz al pin D11 del Arduino Nano.
- Conecta el pin CS de la matriz al pin D10 del Arduino Nano.
- Conecta el pin CLK de la matriz al pin D13 del Arduino Nano

Paso 4: Instalación de la Librería para la Matriz de LEDs

1. Abre el Arduino IDE en tu computadora.
2. Ve a Sketch > Include Library > Manage Libraries.
3. Busca LedControl en el gestor de librerías.
4. Selecciona LedControl y haz clic en Install para añadir la librería a tu entorno de desarrollo.

Paso 5: Configuración del Entorno de Desarrollo

1. Crea un nuevo sketch en el Arduino IDE.
2. Incluye la librería LedControl al inicio del sketch con #include <LedControl.h>.
3. Declara e inicializa la matriz de LEDs

Paso 6: Configuración de la Matriz en el setup()

1. Define la configuración de la matriz de LEDs en la función setup().

Paso 7: Implementación de la Lógica del Temporizador en el loop()

1. Escribe el código necesario en la función loop() para gestionar el temporizador en la matriz de LEDs.

Paso 8: Subida del Código al Arduino Nano

1. Conecta el Arduino Nano a tu computadora usando el cable USB.
2. Selecciona la placa y el puerto correspondientes en el Arduino IDE (Tools > Board > Arduino Nano, y Tools > Port > el puerto correspondiente).
3. Haz clic en el botón Upload para cargar el código al Arduino Nano.
4. Agregar texto

Paso 9: Prueba y Ajustes Finales

1. Desconecta el Arduino Nano de la computadora y conéctalo a la fuente de alimentación.
2. Observa cómo los LEDs se encienden en secuencia representando el flujo de la "arena". Cada minuto, el temporizador se reiniciará automáticamente.

Paso 10: Personalización y Optimización

1. Ajusta el tiempo de la cuenta regresiva cambiando los valores en el código para adaptar la duración del temporizador a tus necesidades.
2. Añade más funcionalidades como sonidos con un buzzer o una interfaz más avanzada.
3. Optimiza el diseño para mejorar la apariencia y funcionalidad de tu reloj de arena digital.

Codigo de programación

Estoy_cansado_jefe

```
#include <LedControl.h>
LedControl lc=LedControl(10,8,9,1);


#define demora 1000


byte cero[8]= {
  B00111000,
  B01000100,
  B01000100,
  B01000100,
  B01000100,
  B01000100,
  B01000100,
  B00111000
};


byte uno[8]= {
  B00010000,
  B00110000,
  B01010000,
  B00010000,
  B00010000,
  B00010000,
  B00010000,
  B00111000
};
```

```
byte tres[8]= {
  B00111000,
  B01000100,
  B00000100,
  B00011000,
  B00000100,
  B00000100,
  B01000100,
  B00111000
};

byte cuatro[8]= {
  B00001000,
  B00011000,
  B00101000,
  B01001000,
  B01001000,
  B01111100,
  B00001000,
  B00001000
```

```
byte cinco[8]= {
  B01111100,
  B01000000,
  B01000000,
  B01111000,
  B00000100,
  B00000100,
  B01000100,
  B00111000
};

byte seis[8]= {
  B00111000,
  B01000100,
  B01000000,
  B01111000,
  B01000100,
  B01000100,
  B01000100,
  B00111000
```

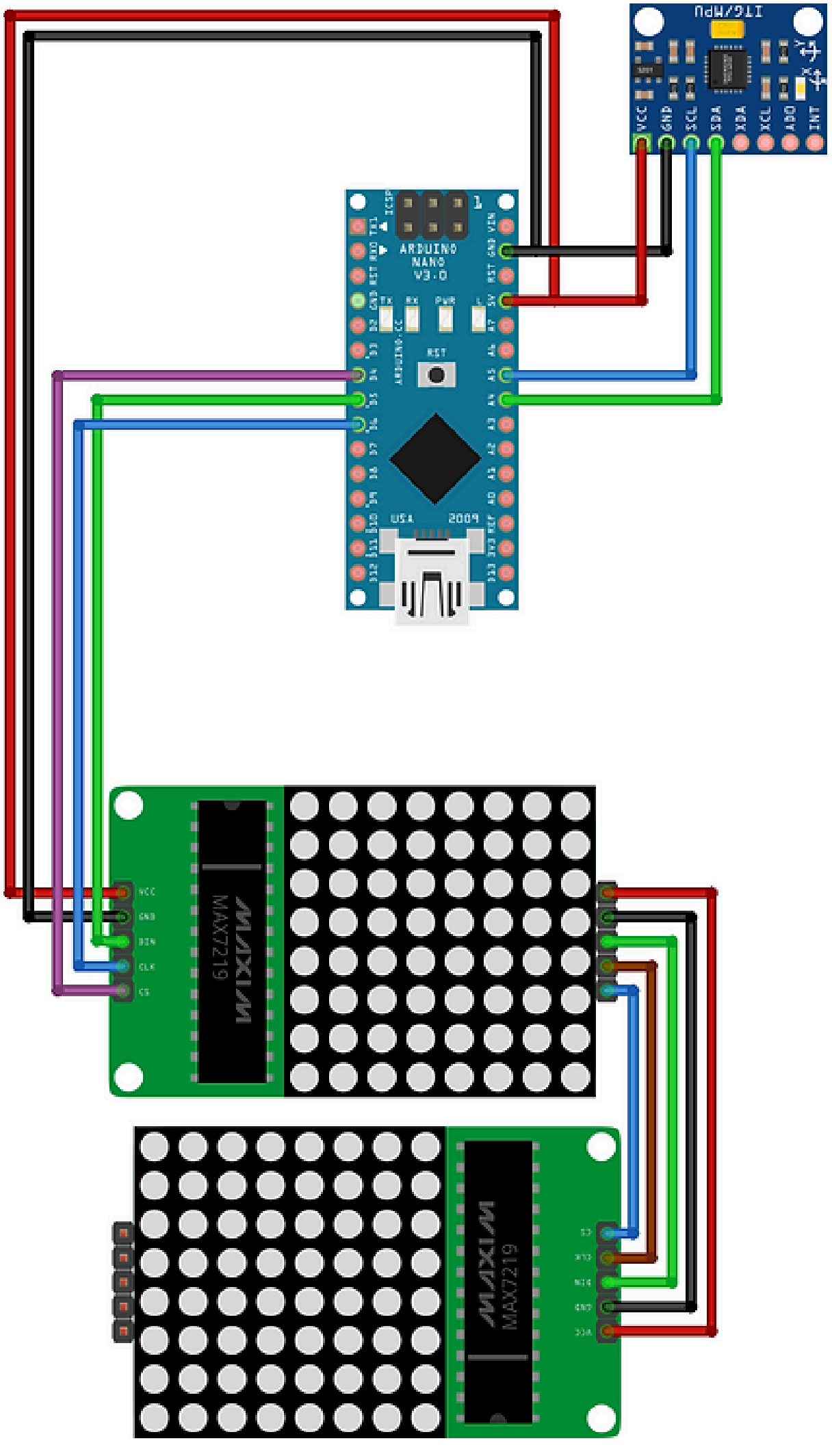
```
byte siete[8]= {
  B01111100,
  B00000100,
  B00000100,
  B00001000,
  B00010000,
  B00100000,
  B00100000,
  B00100000
};

byte ocho[8]= {
  B00111000,
  B01000100,
  B01000100,
  B00111000,
  B01000100,
  B01000100,
  B01000100,
  B00111000
```

```
  B00111000
};

byte nueve[8]= {
  B00111000,
  B01000100,
  B01000100,
  B01000100,
  B00111100,
  B00000100,
  B01000100,
  B00111000
};

void setup() {
  lc.shutdown(0,false);      // enciende la matriz
  lc.setIntensity(0,15);      // establece brillo
  lc.clearDisplay(0);      // blanquea matriz
}

void loop(){
```

```
void loop(){
  mostrar_0();
  delay(demora);
  mostrar_1();
  delay(demora);
  mostrar_2();
  delay(demora);
  mostrar_3();
  delay(demora);
  mostrar_4();
  delay(demora);
  mostrar_5();
  delay(demora);
  mostrar_6();
  delay(demora);
  mostrar_7();
  delay(demora);
  mostrar_8();
  delay(demora);
  mostrar_9();
```

```
void mostrar_0(){
  for (int i = 0; i < 8; i++)
  {
    lc.setRow(0,i,cero[i]);
  }
}

void mostrar_1(){
  for (int i = 0; i < 8; i++)
  {
    lc.setRow(0,i,uno[i]);
  }
}

void mostrar_2(){
  for (int i = 0; i < 8; i++)
  {
    lc.setRow(0,i,dos[i]);
  }
```

```
}


void mostrar_7(){
  for (int i = 0; i < 8; i++)
  {
    lc.setRow(0,i,siete[i]);
  }
}


void mostrar_8(){
  for (int i = 0; i < 8; i++)
  {
    lc.setRow(0,i,ocho[i]);
  }
}


void mostrar_9(){
  for (int i = 0; i < 8; i++)
  {
    lc.setRow(0,i,nueve[i]);
  }
```


ARDUINO
NANO
V3.0
USA
2009
RST
MAX7219
MAX7219