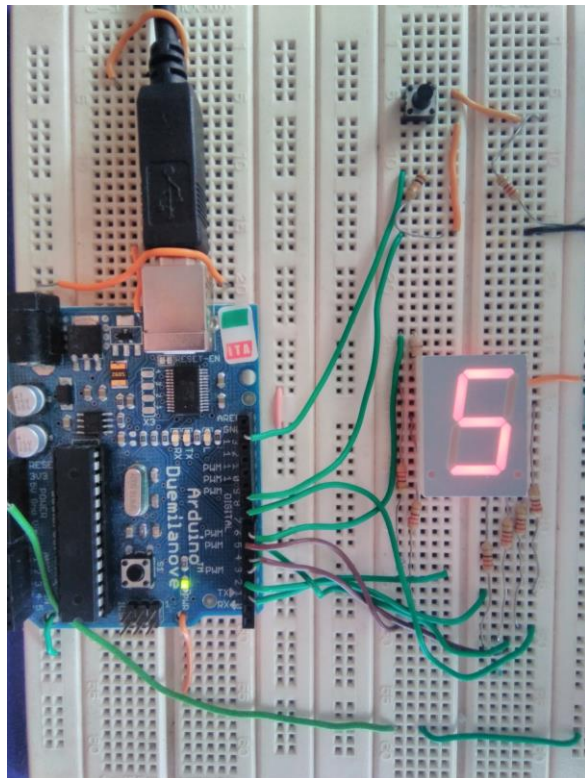


Arduino

**Aplicando aleatoriedad a
un display de 7 segmentos.
(DADO ELECTRÓNICO)**



Introducción

En esta ocasión se ha diseñado un circuito con muy pocos componentes, todos ellos de fácil conseguir, y donde esencialmente se utiliza un microcontrolador IC1 Atmega 328P-PU programado en Arduino que realiza la operación de visualización de un número del 1 al 6, aleatoriamente, cuando se actúa en un pulsador, simulando el típico dado de 6 números para los juegos de azar.

El diseño del circuito eléctrico está formado en cuatro partes principales:

1. La alimentación del circuito.
2. El microcontrolador Atmega 328P-PU con la programación.
3. El pulsador S2 de inicio
4. La conexión del display de 7 segmentos.

La alimentación del circuito

Sencillamente la alimentación para todo el circuito se lleva a cabo mediante un grupo de tres pilas AA en serie consiguiendo un voltaje de 4,5 voltios. Para su conexión y desconexión de la alimentación se conectará un pequeño interruptor S1 que abre o cierra (ON-OFF) el paso de corriente al circuito.

El microcontrolador Atmega 328P-PU

Este microcontrolador por sus excelentes características puede realizar una infinidad de operaciones, tanto de lectura, escritura, temporización, visualización, aleatoriedad, etc., en el entorno de programación de Arduino.

En este caso utilizamos 8 E/S digitales, 7 de las cuales son salidas, D2 a D8, para encender los 7 segmentos del display que se activan a nivel bajo ya que los diodos led se encuentran conectados en ánodo común con +Vcc. D9 se utiliza de entrada digital que se activa con nivel alto cuando se pulsa S2.

El pulsador S2 de inicio

La función del pulsador S2, normalmente abierto, mientras no se pulse no actúa nada, cuando se pulsa realiza la función de inicio de la secuencia de programación enviando un nivel alto al pin 15 de IC1, D9, y se establece un movimiento de visualización de todos los números durante un breve tiempo, hasta que se detiene en uno de ellos, quedando a la espera de volver a pulsar S2.

La conexión del display de 7 segmentos

El display de 7 segmentos que utilizamos en nuestro diseño se encarga de visualizar un número del 1 al 6 según el programa establecido en el microcontrolador IC1. Al utilizarse un display de 7 segmentos en ánodo común, es decir, sus ánodos están todos conectados internamente al potencial positivo, +Vcc, para que se encienda el segmento LED del display hay que aplicar una tensión negativa de 0 voltios en el cátodo mediante una resistencia limitadora de 330Ω , ésta tensión negativa o 0 voltios es la que proviene de la salida digital del microcontrolador IC1 cuando aplicamos en la programación el nivel LOW.

En la siguiente tabla se describe la conexión de los pines del microcontrolador IC1 con los diferentes segmentos del display.

IC1	SALIDA DIGITAL	DISPLAY
Pin 4	D2	a
Pin 5	D3	b
Pin 6	D4	c
Pin 11	D5	d
Pin 12	D6	e
Pin 13	D7	f
Pin 14	D8	g

Componentes electrónicos

En este circuito se utilizan los siguientes componentes electrónicos:

R1 a R8 = Resistencias de $\frac{1}{4}$ W de 330Ω

R9 = Resistencia $\frac{1}{4}$ W de 10K

C1 y C2 = Condensadores de disco cerámico de 22pF

XT1 = Cristal de cuarzo de 16MHz

IC1= Microcontrolador Atmega328P-PU DIL 28 pines

ZC1 = Zócalo para IC1 de 28 pines.

DLP1 = Display de 7 segmentos, en ánodo común,

S1 = Microinterruptor simple de un circuito.

S2 = Micropulsador simple NA, Normalmente Abierto,

BT1 = Portapilas para 3 pilas del tipo LR6-AA de 1,5 voltios.

Programación en Arduino

En la programación de Arduino utilizada para nuestro diseño, se utiliza la función Random (aleatorio) que genera números aleatorios en un rango de 0 a un máximo, en nuestro caso es un máximo de 6 números: 1, 2, 3, 4, 5 y 6, el rango sería de 0 a 6.

A continuación se describe todas las instrucciones y funciones de la programación en Arduino para ésta aplicación.

```
/*Programa para un número aleatorio del 1 a 6 visualizado en un display de 7
segmentos*/
int estadopulsador; //declaramos variable estadopulsador
int maximo=6; // declaramos variable maximo a 6
int randomNumber; // declaramos variable para numero aleatorio

void setup() {
  pinMode (2, OUTPUT); //ponemos de salida digital el pin 2 letra a
  pinMode (3, OUTPUT); //ponemos de salida digital el pin 3 letra b
  pinMode (4, OUTPUT); //ponemos de salida digital el pin 4 letra c
  pinMode (5, OUTPUT); //ponemos de salida digital el pin 5 letra d
  pinMode (6, OUTPUT); //ponemos de salida digital el pin 6 letra e
  pinMode (7, OUTPUT); //ponemos de salida digital el pin 7 letra f
  pinMode (8, OUTPUT); //ponemos de salida digital el pin 8 letra g
  pinMode (9, INPUT); //ponemos de entrada digital el pulsador pin 9
}
void loop() {
  estadopulsador=digitalRead(9); //leemos el pulsador en el pin 9
  if (estadopulsador==HIGH) //condicional si pulsador nivel alto ejecuta
  {
    for (int w=0; w<4; w++){ //contador de visualizacion de los numeros
      dado1(); //damos de alta una funcion para el numero 1
      dado2(); //damos de alta una funcion para el numero 2
      dado3(); //damos de alta una funcion para el numero 3
      dado4(); //damos de alta una funcion para el numero 4
      dado5(); //damos de alta una funcion para el numero 5
      dado6(); //damos de alta una funcion para el numero 6
    }
    randomSeed(millis()); //genera numeros aleatorios a partir del tiempo en
milisegundos
    randomNumber = random (maximo); //variable que consigue un numero aleatorio a
partir del valor 0 hasta un maximo
    if (randomNumber==1) { //condicional si es 1 activa dado1
      dado1();}
    if (randomNumber==2) { //condicional si es 2 activa dado2
      dado2();}
    if (randomNumber==3) { //condicional si es 3 activa dado3
      dado3();}
    if (randomNumber==4) { //condicional si es 4 activa dado4
      dado4();}
    if (randomNumber==5) { //condicional si es 5 activa dado5
      dado5();}
    if (randomNumber==6) { //condicional si es 6 activa dado6
      dado6();}
  } } /* Importante: En este caso se ha utilizado un display de 7 segmentos en
anodo comun, para ello se envia un nivel bajo o 0 voltios para encender el
segmento correspondiente. En el caso de que sea un display de 7 segmentos
```

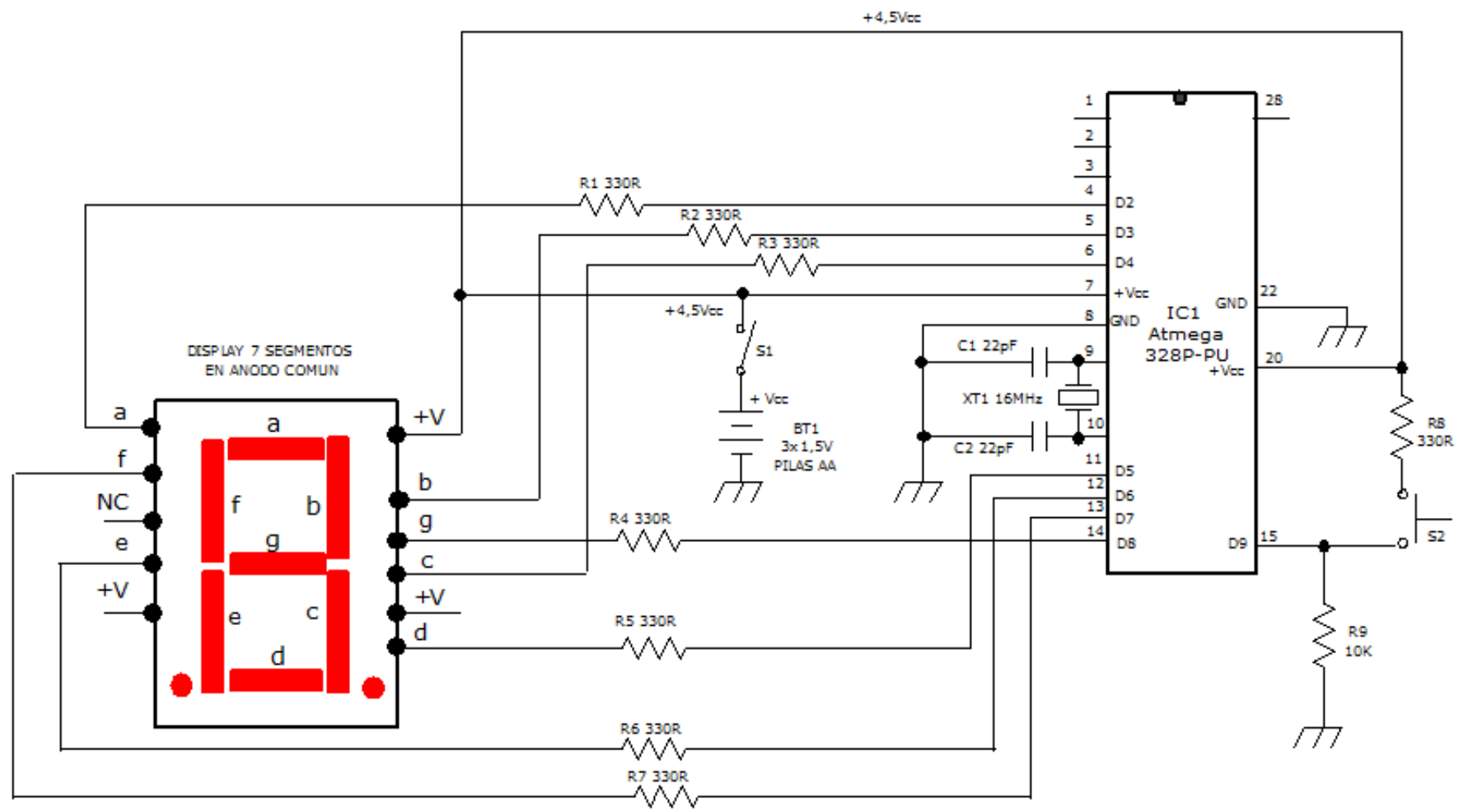
en catodo comun tenemos que cambiar la salida a nivel alto o 5 voltios */

```
void dado1(){ //funcion visualizar numero 1
  digitalWrite(2, HIGH); //pone a nivel alto el pin 2
  digitalWrite(3, LOW); //pone a nivel bajo el pin 3
  digitalWrite(4, LOW); //pone a nivel bajo el pin 4
  digitalWrite(5, HIGH); //pone a nivel alto el pin 5
  digitalWrite(6, HIGH); //pone a nivel alto el pin 6
  digitalWrite(7, HIGH); //pone a nivel alto el pin 7
  digitalWrite (8,HIGH); //pone a nivel alto el pin 8
  delay (100); // retarda 100ms
}
void dado2(){ //funcion visualizar numero 2
  digitalWrite(2, LOW); //pone a nivel bajo el pin 2
  digitalWrite(3, LOW); //pone a nivel bajo el pin 3
  digitalWrite(4, HIGH); //pone a nivel alto el pin 4
  digitalWrite(5, LOW); //pone a nivel bajo el pin 5
  digitalWrite(6, LOW); //pone a nivel bajo el pin 6
  digitalWrite(7, HIGH); //pone a nivel alto el pin 7
  digitalWrite (8,LOW); //pone a nivel bajo el pin 8
  delay (100); //retardo de 100ms
}
void dado3(){ //funcion visualizar numero 3
  digitalWrite(2, LOW); //pone a nivel bajo el pin 2
  digitalWrite(3, LOW); //pone a nivel bajo el pin 3
  digitalWrite(4, LOW); //pone a nivel bajo el pin 4
  digitalWrite(5, LOW); //pone a nivel bajo el pin 5
  digitalWrite(6, HIGH); //pone a nivel alto el pin 6
  digitalWrite(7, HIGH); //pone a nivel alto el pin 7
  digitalWrite (8,LOW); //pone a nivel bajo el pin 8
  delay (100); //retardo de 100ms
}
void dado4(){ //funcion visualizar numero 4
  digitalWrite(2, HIGH); //pone a nivel alto el pin 2
  digitalWrite(3, LOW); //pone a nivel bajo el pin 3
  digitalWrite(4, LOW); //pone a nivel bajo el pin 4
  digitalWrite(5, HIGH); //pone a nivel alto el pin 5
  digitalWrite(6, HIGH); //pone a nivel alto el pin 6
  digitalWrite(7, LOW); //pone a nivel bajo el pin 7
  digitalWrite (8,LOW); //pone a nivel bajo el pin 8
  delay (100); //retardo de 100ms
}
void dado5(){ //funcion visualizar numero 5
  digitalWrite(2, LOW); //pone a nivel bajo el pin 2
  digitalWrite(3, HIGH); //pone a nivel alto el pin 3
  digitalWrite(4, LOW); //pone a nivel bajo el pin 4
  digitalWrite(5, LOW); //pone a nivel bajo el pin 5
  digitalWrite(6, HIGH); //pone a nivel alto el pin 6
  digitalWrite(7, LOW); //pone a nivel bajo el pin 7
  digitalWrite (8,LOW); //pone a nivel bajo el pin 8
  delay (100); //retardo de 100ms
}
void dado6(){ //funcion visualizar numero 6
  digitalWrite(2, LOW); //pone a nivel bajo el pin 2
  digitalWrite(3, HIGH); //pone a nivel alto el pin 3
  digitalWrite(4, LOW); //pone a nivel bajo el pin 4
  digitalWrite(5, LOW); //pone a nivel bajo el pin 5
  digitalWrite(6, LOW); //pone a nivel bajo el pin 6
  digitalWrite(7, LOW); //pone a nivel bajo el pin 7
  digitalWrite (8,LOW); //pone a nivel bajo el pin 8
  delay (100); //retardo de 100ms }
```

Esquema eléctrico

A continuación se muestra el plano eléctrico de conexiones de los diferentes componentes electrónicos que se utilizan.

Con todo ello, una vez programado el microcontrolador en la placa de Arduino se puede desmontar con mucho cuidado y volverlo a montar en una placa multitaladros y en un zócalo de 28 pines y tal como se describe en el plano eléctrico se conectan los demás componentes: resistencias, condensadores, cristal de cuarzo, portapilas, pulsador y display de 7 segmentos. Introduciendo el circuito en una cajita de plástico mecanizada para su protección y manipulación.



APLICANDO ALEATORIEDAD A UN DISPLAY DE 7 SEGMENTOS (DADO ELECTRONICO)

Plano:	Fecha: 30/08/2022	Nº de Hojas: 1/1
P- 2288	Dibujado: Jose M. Castillo Castillo	