

QUINIELA ELECTRÓNICA

Introducción

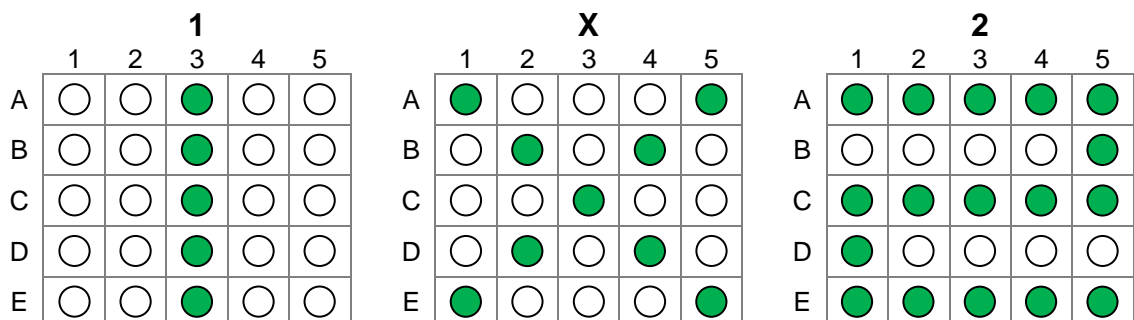
Este circuito sirve para generar aleatoriamente los posibles resultados de una quiniela (1, X y 2). Se basa en generar un número aleatorio mediante un PICAXE 08, comprendido entre 0 y 255 y a partir de él, mostrar uno de los tres símbolos de la quiniela en una matriz de 25 diodos led. A la hora de programar el circuito, hay que tener en cuenta la probabilidad que tiene de salir cada símbolo. El uno tiene el 50%, la X el 28% y el 2 el 22%.

Componentes

Descripción	Cantidad	Precio aprox. €
Led 5 mm de diámetro	25	0.10
Diodo 1N4148	7	0.05
Resistor entre 220 Ω ¼ w	23	0.02
Transistor BC107	3	0.10
Pulsador miniatura rosca	1	0.50
Placa perforada circuito impreso tiras	1	3
Placa PICAXE08	1	-

Diseño de la matriz de diodos led

El problema que aparece a la hora de diseñar el display es que hay que controlar 23 led usando tres de las cuatro salidas de un Picaxe 08. Para abordar el problema, comenzamos analizando la forma en que se van a mostrar los distintos caracteres:



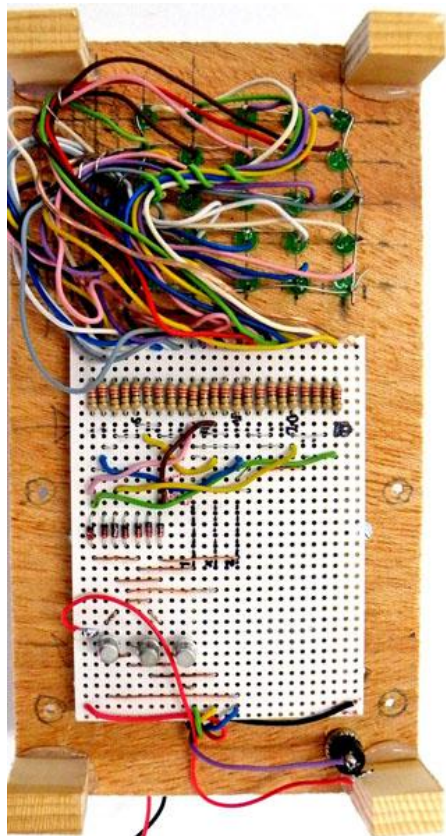
De este esquema se puede sacar la siguiente información:

■ Combinaciones para cada dígito:

1: A3, B3, C3, D3, E3

X: A1, A5, B2, B4, C3, D2, D4, E1, E5

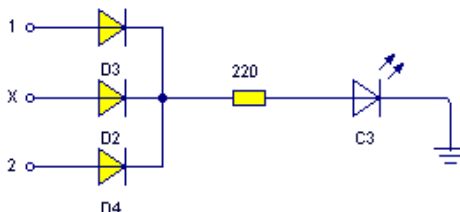
2: A1, A2, A3, A4, A5, B5, A1, C2, C3, C4, C5, D1, E1, E2, E3, E4, E5



■ Combinaciones comunes a varios dígitos:

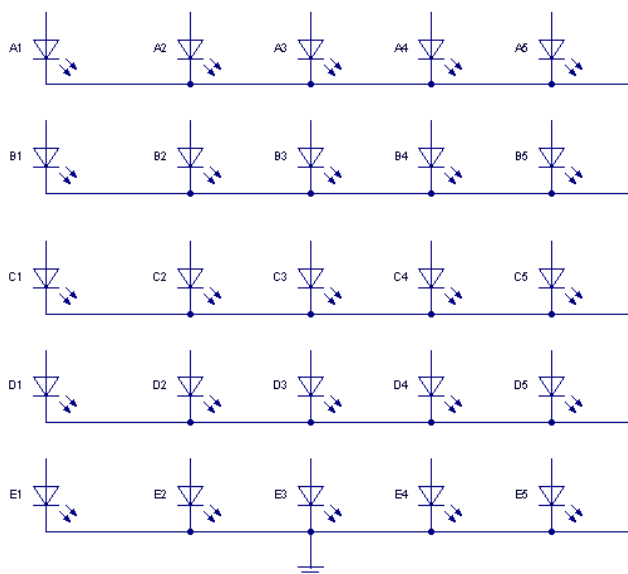
C3	1, X, 2
A1	X, 2
A5	X, 2
E1	X, 2
E5	X, 2
A3	1, 2
E3	1, 2

Estas combinaciones nos presentan el siguiente problema: al ser comunes a varios dígitos, si estos están formados por un grupo de leds en paralelo, por él o los leds comunes pasa la corriente de un grupo a otro, de modo que al conectar una de las combinaciones también se enciende otra con la que tiene diodos comunes. Para solucionar esto, tenemos que intercalar diodos en las conexiones de estos leds de modo que no pueda circular la corriente hacia atrás y alimentar a otro circuito. Como ejemplo, veamos como sería el circuito para el led central (C3):

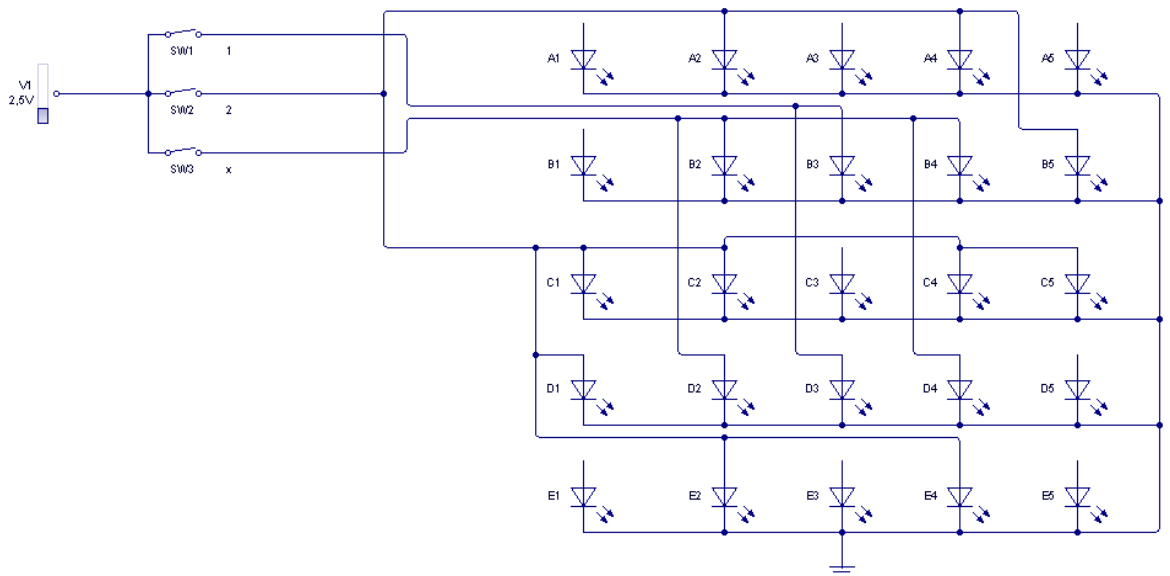


Los diodos colocados impiden que al poner a tensión el terminal x, llegue ésta al terminal 1 o 2.

Antes de montar el circuito de la matriz de diodos lo simulamos con Livewire y buscamos posibles simplificaciones. El primer paso para simular el circuito es montar la matriz de diodos, de forma similar a la que se muestra aquí abajo, con todos los cátodos de los diodos unidos y puestos a masa:

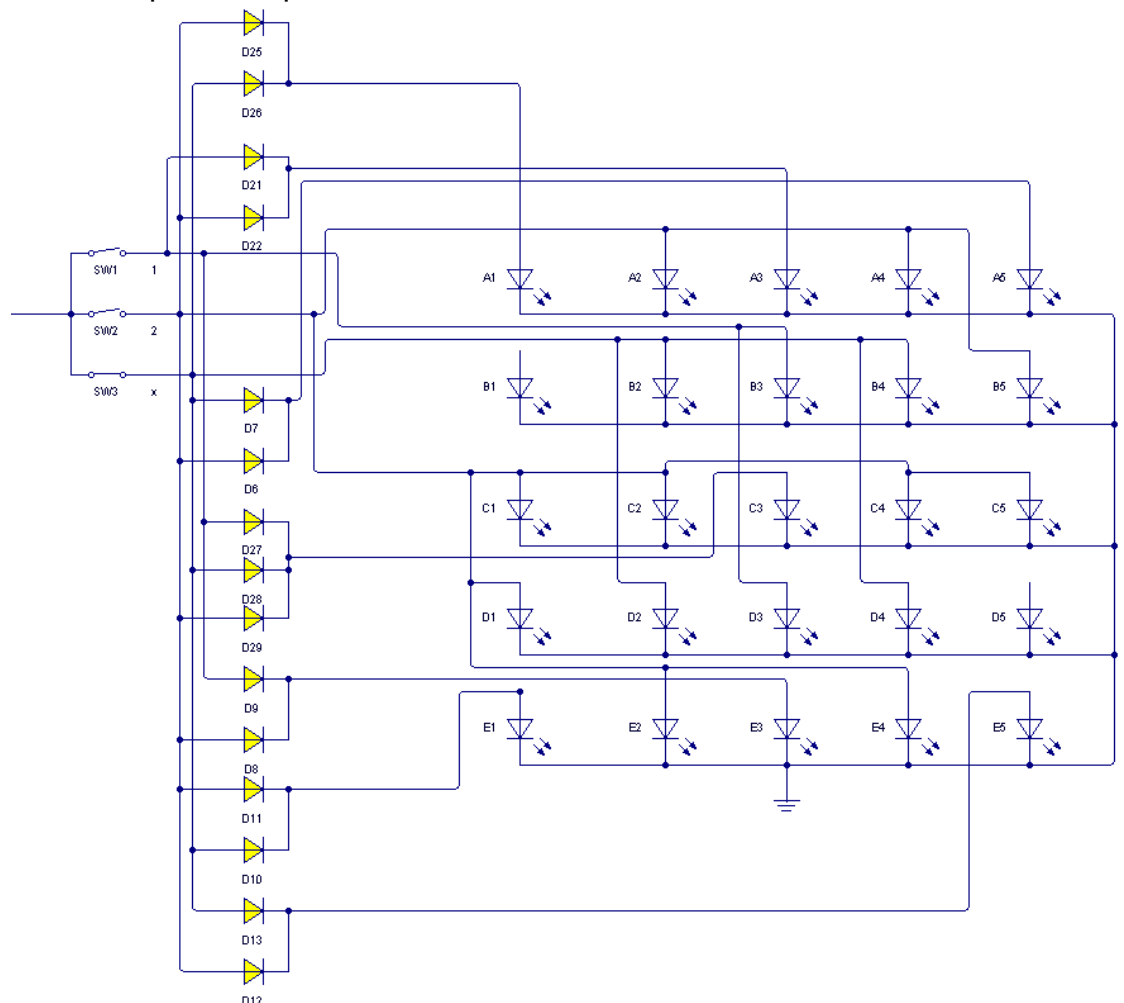


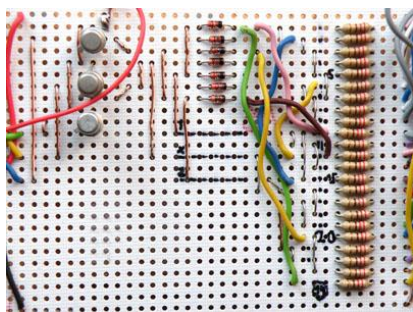
Ahora, vamos a ir conectando los ánodos de los diodos que solo pertenecen a un dígito:



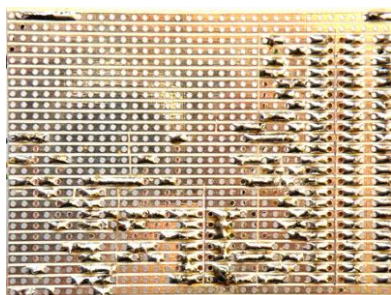
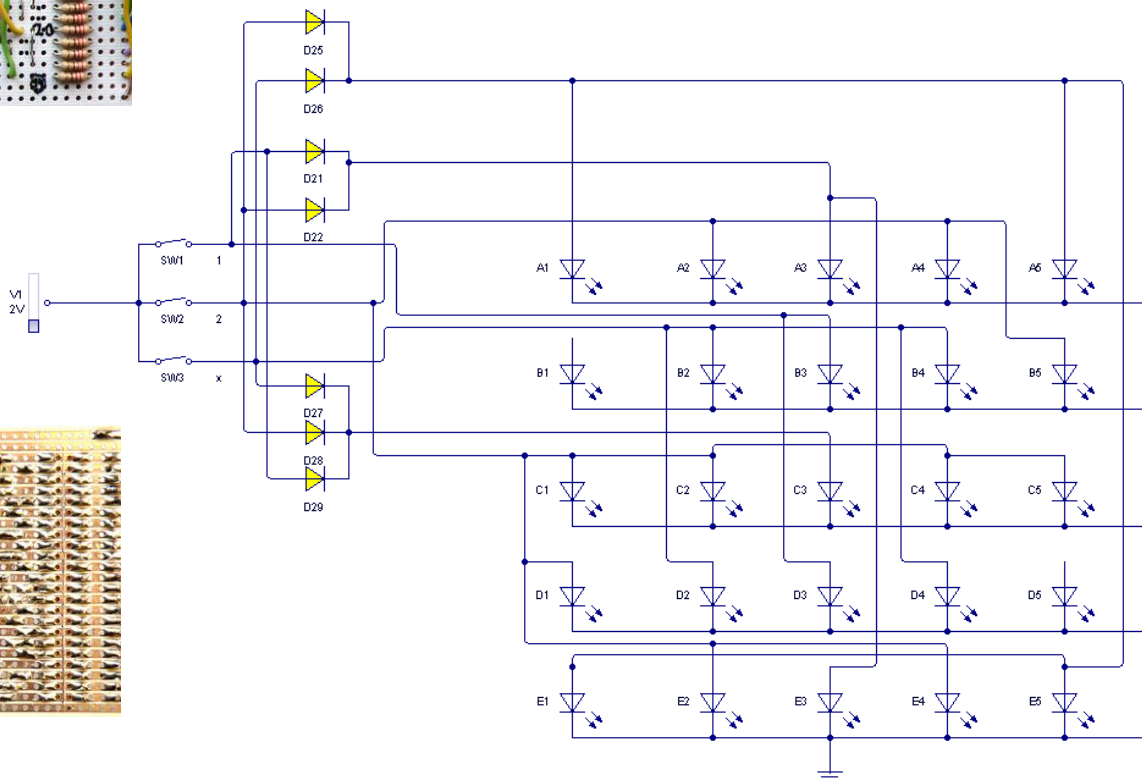
Ahora vamos a conectar los leds que nos quedan, que son comunes a varios dígitos. En este caso tenemos que usar diodos para independizar las salidas:

LED	Dígitos a los que pertenece
A1	X, 2
A2	2
A3	1, 2
A4	2
A5	X, 2
B1	Nada
B2	X
B3	1
B4	X
B5	2
C1	2
C2	2
C3	1, X, 2
C4	2
C5	2
D1	2
D2	X
D3	1
D4	X
D5	Nada
E1	X, 2
E2	2
E3	1, 2
E4	2
E5	2, X





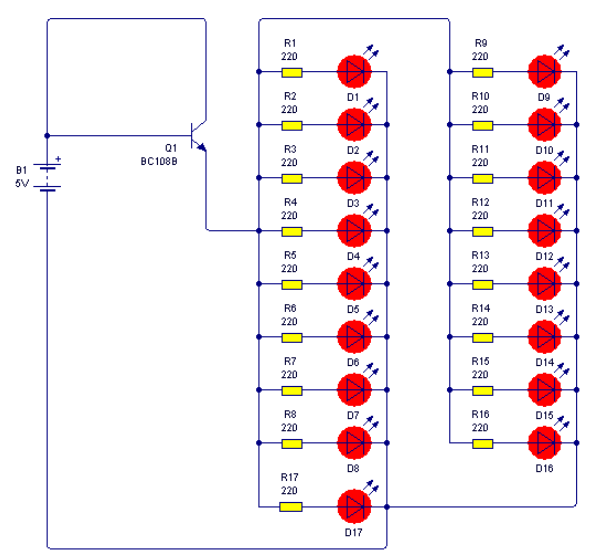
Si observamos el circuito vemos que hay varios leds que se activan por la misma combinación de entradas, por ejemplo, A1, A5, E1 y E5 se activan mediante las entradas x o 2. Pues bien, podemos simplificar el circuito poniendo todos estos leds en paralelo y conectados a la salida de una misma pareja de diodos.



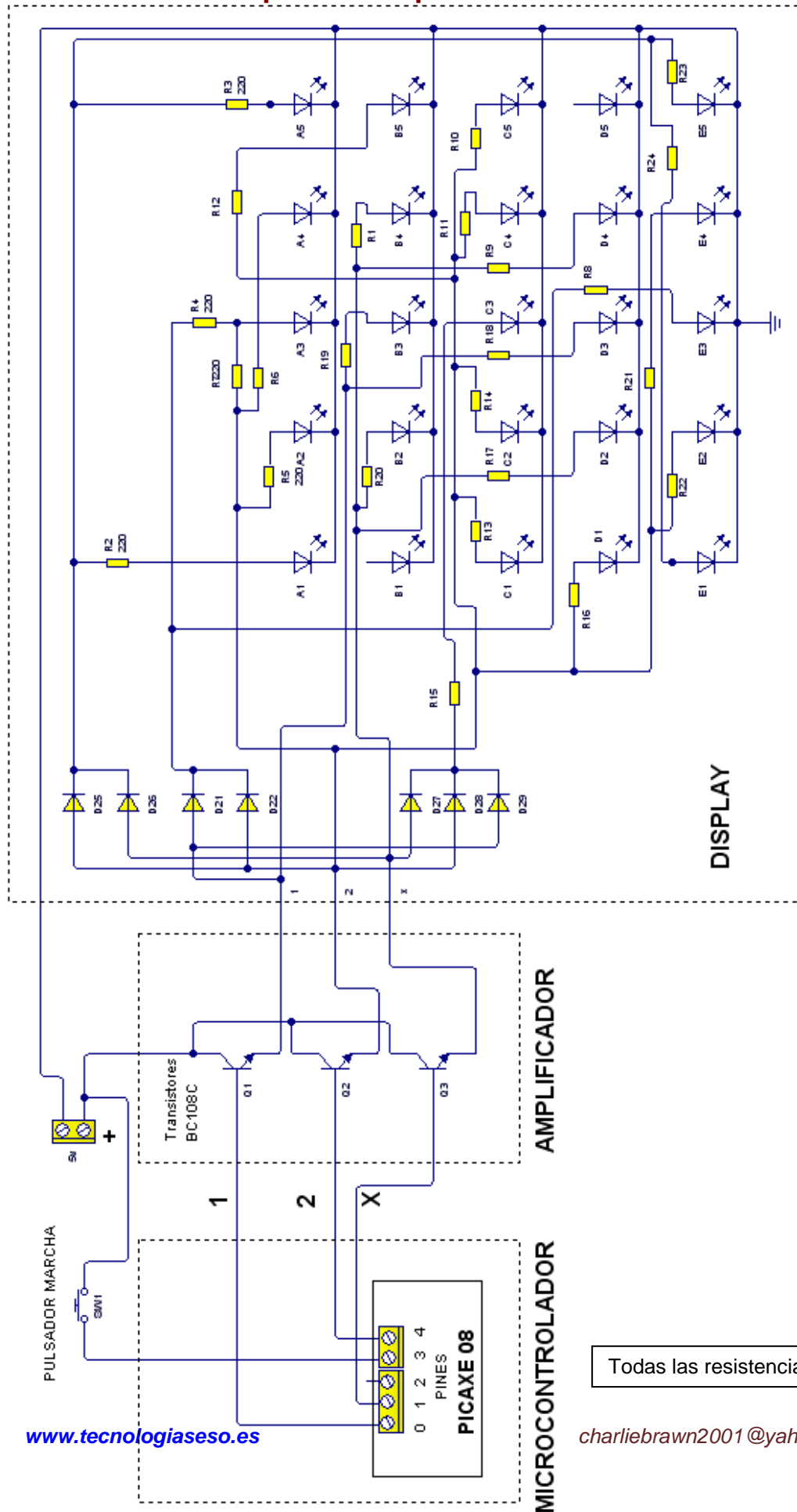
Para montar el circuito tenemos que tener en cuenta dos cosas: primero, que los leds se van a alimentar a 5 v, por lo que cada uno ha de llevar una resistencia en serie de 220Ω , $\frac{1}{4}$ w, y segundo, que la salida del microcontrolador no es capaz de alimentar un conjunto de leds tan grande, por lo que habrá que intercalar un transistor o amplificador de algún tipo. Teniendo en cuenta lo dicho, el circuito para el dígito que más leds posee (el 2), quedará así:

El transistor BC108 soporta una corriente máxima de 100 ma. En el caso más desfavorable nuestro circuito consume una intensidad de 171 ma. Esto no supone un problema debido a que el tiempo que va a estar suministrando esa intensidad es muy pequeño y en la practica la intensidad es algo menor a la obtenida en el simulador. Al estar el transistor totalmente saturado no será necesario colocar una resistencia de base

Valores medidos con Livewire		
I_{led}	10,1	mA
I_{total}	171	mA
I_b	1,11	mA
V_{ce}	897	mV



Esquema completo:



Programación

```
*****
*****      QUINIELA      *****
*****
```

```
*****CONEXIONES*****
```

```
'PULSADOR AL PIN 3
'SALIDA 1 AL PIN 0
'SALIDA X AL PIN 1
'SALIDA 2 AL PIN 4
```

Inicio:

```
low 0          'pone las salidas que generan los dígitos a nivel
                bajo
```

```
low 1
low 4
```

```
If pin3 is on then aleatorio    'Si el pin 1 esta a nivel alto ejecuta la
                                'subrutina "aleatorio
```

```
goto inicio          'si no se cumple la condición anterior
                    'vuelve a inicio
```

```
aleatorio:
random b1            'Genera un número pseudoaleatorio y lo
                    'guarda el b1
```

creadigito:

```
if b1>200 then dos    'si el numero aleatorio guardado en b1 (de 0 a
                    '255) es mayor que 200, entonces salta a la
                    'subrutina dos
```

```
if b1>128 then equis  'si el numero aleatorio guardado en b1 es mayor
                    'que 128, entonces salta a la subrutina equis
```

```
uno:                'esta subrutina se ejecuta si no se cumplen las
                    'dos condiciones anteriores, es decir, si el número
                    'almacenado en b1 es menor a 128
```

```
high 0              'pone la salida cero (que genera un uno en la
                    'pantalla a nivel alto
wait 1              'espera un segundo
goto inicio         'saltar a inicio
```

```
equis:             'igual que la anterior pero para la x
high 1
wait 1
goto inicio
```

```
dos:               'igual que la anterior pero para el 2
high 4
wait 1
goto inicio
```

■ **Prueba:**

Si realizamos un número elevado de tiradas vemos que se repite la siguiente secuencia:

1 1 1 1 X 1 2 2 2 X 1 X

Esto es debido a que el microcontrolador genera un número pseudoaleatorio. Para solventar este problema podemos modificar el inicio del programa de esta forma:

inicio:

low 0
low 1
low 4

random b1

if input3 is on then aleatorio
goto quiniela

aleatorio:

if b1>200 then dos
if b1>128 then equis

uno:

high 0
wait 1
low 0
goto inicio

dos:

high 4
wait 1
low 4
goto inicio

equis:

high 1
wait 1
low 1
goto inicio

El programa está generando continuamente números aleatorios (miles cada segundo). Al accionar el pulsador conectado al pin 3, se detiene este bucle y el número aleatorio que hay almacenado en b1 en ese instante se utiliza para generar la salida

■ **Mejora del programa:**

Se puede mejorar el programa añadiendo efectos visuales que hagan que parezca que el resultado es el final de una secuencia de dígitos que va desacelerando hasta que para. La secuencia podría ser algo así:

high 0	}	Se pone a nivel alto la salida 0 (se muestra un 1) durante 20 ms y luego se apaga
pause 20		
low 0		
high 1	}	Se pone a nivel alto la salida 1 (se muestra una x) durante 50 ms y luego se apaga
pause 50		
low 1		
high 4	}	Se pone a nivel alto la salida 4 (se muestra un 2) durante 75 ms y luego se apaga
pause 75		
low 4		
high 0	}	Se repite la secuencia aumentando el tiempo que se mantiene iluminado cada símbolo
pause 100		
low 0		
high 1		
pause 150		
low 1		
high 4		
pause 200		
low 4		

Este bloque lo intercalamos antes de la subrutina “uno”, de modo que se muestre siempre antes de salir un dígito.

Por último, en las subrutinas “equis” y “dos” podemos continuar la secuencia para que cuadre, o sea, que siga la pauta 1, X, 2, 1, X.... Como el bloque termina en 2, viene bien para la subrutina “uno”, pero para las “equis” hay que hacer que se muestre un 1 y espere unos 250 ms antes de que aparezca este símbolo.