

```
/* TEST TECLADO NUMÉRICO */
```

```
//El test pretende corroborar el teclado numérico envía correctamente la información al
//pulsar las teclas que contiene. Es un teclado matricial de 12 teclas distribuidas tal y
//como indica en el KEYMAP descrito en el programa.
//Este ejemplo se ha realizado para un teclado matricial de 12 teclas, y una placa Arduino.
//ATMega que contiene un número mayor de pines de conexión. Además incluye la librería que
//aparece a continuación, y que está diseñada para teclados numéricos matriciales como el
//que nosotros utilizamos.
```

```
#include <Keypad.h>
```

```
//Definimos un vector de 25 caracteres que va a contener los valores de las teclas que se
//hayan pulsado
char vector[25]={0};
```

```
int i=0; // inicializamos el contador
```

```
const byte ROWS = 4;           //definimos que nuestro teclado va a tener cuatro filas
const byte COLS = 3;           //definimos que nuestro teclado va a tener tres columnas
```

```
//Definimos el KEYMAP, es decir cual es la posición matricial de nuestro teclado
char keys[ROWS][COLS] = {
  {1,2,3},
  {4,5,6},
  {7,8,9},
  {*,0,#}
};
```

```
//Conexión del keypad ROW0, ROW1, ROW2 y ROW3 a estos pines de Arduino.
```

```
byte rowPins[ROWS] = { 41, 51, 49, 45 };
```

```
//Conexión del keypad COL0, COL1 and COL2 a estos pines de Arduino.
```

```
byte colPins[COLS] = { 43, 39, 47 };
```

```
//Creamos el Keypad llamando a la siguiente función y con los valores antes asignados
Keypad kpd = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
```

```
void setup()
{
  Serial.begin(9600);           // Inicializa la comunicacion serie
}
```

```
void loop()
{
  // Lectura de datos del keyboard (teclado)
  int key = kpd.getKey();

  // Ahora vamos hacer que se guarden y se escriban en pantalla
  if(key) // es como if(key != NO_KEY)
  {
    switch (key)
    {
      case 1:
        vector[i]=key;           // Introducimos el valor de la pulsación en la posición i del vector
        Serial.print(vector[i]); // Mostramos el valor de esa pulsación en pantalla
        i++;                     // Sumamos 1 al puntero para que apunte a la siguiente posición del vector
        break;
    }
  }
}
```

```
case 2:
    vector[i]=key;    // Introducimos el valor de la pulsación en la posición i del vector
    Serial.print(vector[i]); // Mostramos el valor de esa pulsación en pantalla
    i++;    // Sumamos 1 al puntero para que apunte a la siguiente posición del vector
break;
```

```
case 3:
    vector[i]=key;    // Introducimos el valor de la pulsación en la posición i del vector
    Serial.print(vector[i]); // Mostramos el valor de esa pulsación en pantalla
    i++;    // Sumamos 1 al puntero para que apunte a la siguiente posición del vector
break;
```

```
case 4:
    vector[i]=key;    // Introducimos el valor de la pulsación en la posición i del vector
    Serial.print(vector[i]); // Mostramos el valor de esa pulsación en pantalla
    i++;    // Sumamos 1 al puntero para que apunte a la siguiente posición del vector
break;
```

```
case 5:
    vector[i]=key;    // Introducimos el valor de la pulsación en la posición i del vector
    Serial.print(vector[i]); // Mostramos el valor de esa pulsación en pantalla
    i++;    // Sumamos 1 al puntero para que apunte a la siguiente posición del vector
break;
```

```
case 6:
    vector[i]=key;    // Introducimos el valor de la pulsación en la posición i del vector
    Serial.print(vector[i]); // Mostramos el valor de esa pulsación en pantalla
    i++;    // Sumamos 1 al puntero para que apunte a la siguiente posición del vector
break;
```

```
case 7:
    vector[i]=key;    // Introducimos el valor de la pulsación en la posición i del vector
    Serial.print(vector[i]); // Mostramos el valor de esa pulsación en pantalla
    i++;    // Sumamos 1 al puntero para que apunte a la siguiente posición del vector
break;
```

```
case 8:
    vector[i]=key;    // Introducimos el valor de la pulsación en la posición i del vector
    Serial.print(vector[i]); // Mostramos el valor de esa pulsación en pantalla
    i++;    // Sumamos 1 al puntero para que apunte a la siguiente posición del vector
break;
```

```
case 9:
    vector[i]=key;    // Introducimos el valor de la pulsación en la posición i del vector
    Serial.print(vector[i]); // Mostramos el valor de esa pulsación en pantalla
    i++;    // Sumamos 1 al puntero para que apunte a la siguiente posición del vector
break;
```

```
case *:
    vector[i]=key;    // Introducimos el valor de la pulsación en la posición i del vector
    Serial.print(vector[i]); // Mostramos el valor de esa pulsación en pantalla
    i++;    // Sumamos 1 al puntero para que apunte a la siguiente posición del vector
break;
```

```
case #:
    vector[i]=key;    // Introducimos el valor de la pulsación en la posición i del vector
    Serial.print(vector[i]); // Mostramos el valor de esa pulsación en pantalla
    i++;    // Sumamos 1 al puntero para que apunte a la siguiente posición del vector
break;
```

```
}
}
}
```