

```
/* Bee-Bot UPNA */
```

```
//Este ejemplo es un programa para arduino que utiliza las librerías de arduino  
//que aparecen a continuación, y que están diseñadas para la utilización de una  
//pantalla LCD de 20x4 y un teclado matricial de 12 teclas.
```

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

```
//Para este caso utilizamos la placa Arduino ATmega que tiene más pines de  
//conexión ya que necesitamos un número mayor para controlar la pantalla LCD, el  
//teclado numérico y los motores DC.
```

```
//Pretendemos que nuestro robot realice una serie de movimientos que se le abrán  
//indicado en el teclado matricial para después realizarlos todos seguidos. Tras  
//asignar todos los pines de los componentes a los pines de la placa vamos a  
//separar el programa en dos partes.
```

```
//La primera parte va a constar en la fase de recogida de datos, es decir que  
//cuando se vayan introduciendo los valores de direcciones (2, 4, 6 y 8) se  
//vayan guardando, y cuando se pulse el botón de START (5) comience el  
//programa. También hemos incluido las acciones de CLEAR (7) y la de mostrar el  
//vector de movimiento (9).
```

```
//En la segunda fase del programa va realizando la lectura del vector de  
//movimientos y va indicando a los motores los movimientos que tiene que  
//realizar.
```

```
//Los pines de conexión de la pantalla LCD con la placa Arduino son los valores  
//pares de los pines 38 a 48  
LiquidCrystal lcd( 38, 40, 42, 44, 46, 48);
```

```
//Iniciamos una serie de variables  
int vector[10]={0}; //definimos el número de entradas de nuestro vector  
int i=0; //inicializamos el puntero i  
int k=0; //inicializamos el puntero k  
int N=10; //inicializamos una constante que el tamaño del vector de movimientos
```

```
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}  
};
```

```
//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 );
```

```
//Definimos los pines motores que van a controlar la velocidad y la dirección  
int dirA = 11;  
int dirB = 10;  
int speedA = 13;  
int speedB = 12;
```

```

//Definimos los pines de los LEDs que nos van a ofrecer información del estado
//del programa
int LED1 = 8;
int LED2 = 9;

// Ahora iniciamos la parte de configuración

void setup()
{
  Serial.begin(9600); // Inicializa la comunicacion serie
  lcd.begin(20, 4); // Definimos el número de filas y columnas de la pantalla
  pinMode (dirA, OUTPUT); // Ponemos el pin de la dirección A como salida
  pinMode (dirB, OUTPUT); // Ponemos el pin de la dirección B como salida
  pinMode (speedA, OUTPUT); // Ponemos el pin de la velocidad A como salida
  pinMode (speedB, OUTPUT); // Ponemos el pin de la velocidad B como salida
  pinMode (LED1, OUTPUT); // Ponemos el pin del LED1 como salida
  pinMode (LED2, OUTPUT); // Ponemos el pin del LED2 como salida
}

//Comienza el programa que va a separarse en dos fases, la fase de escritura
//(pulsación más grabación de datos) que incluye todos los casos del bucle
//principal switch; y la fase de lectura que comienza y acaba en el caso 5 y
//que es donde realiza todos los movimientos previamente definidos en la parte
//de escritura.

void loop() {

  Serial.println("Comienza");
  lcd.setCursor(0, 0); // Apuntamos a la posición (1,1) de la pantalla
  lcd.print("BEE-BOT ARDUINO UPNA"); // Saca esa frase en pantalla

  // parte de lectura de datos del keyboard (teclado)
  int key = kpd.getKey();

  if(key) // lo mismo que if(key != NO_KEY)
  {
    switch (key)
    {
      //Caso de pulsación hacia ARRIBA
      case 2:
        vector[i]=key; // Introducimos el valor de la pulsación en la posición
                        // i del vector
        i++; // Sumamos 1 al puntero para que apunte a la siguiente posición
              // del vector
        break;

      //Caso de pulsación hacia la IZQUIERDA
      case 4:
        vector[i]=key;
        i++;
        break;

      //Caso de pulsación hacia la DERECHA
      case 6:
        vector[i]=key;
        i++;
        break;
    }
  }
}

```

```
//Caso de pulsación hacia ATRÁS
```

```
case 8:  
    vector[i]=key;  
    i++;  
break;
```

```
//Caso de pulsación para dar comienzo al movimiento
```

```
case 5:
```

```
Serial.println("START"); // Mostramos la palabra comienzo  
digitalWrite(LED1, HIGH); // Encendemos el LED1 para comenzar  
                           // el movimiento  
lcd.setCursor(7, 1); //Ponemos el puntero en la 2ª línea  
lcd.print("START ");  
delay(100);  
digitalWrite(LED1, LOW);  
k=0; // Inicializamos el contador del siguiente bucle para leer  
     // el vector
```

```
do {
```

```
switch (vector[k]) {  
    case 2: // Realiza un paso hacia adelante  
        Serial.println("adelante");  
        digitalWrite (dirA, HIGH);  
        digitalWrite (dirB, HIGH);  
        analogWrite (speedA, 245);  
        analogWrite (speedB, 250);  
        lcd.setCursor(7, 1);  
        lcd.print(" UP ");  
        delay(400); // realiza el movimiento durante 1 segundo  
        Serial.println("parada");  
        digitalWrite (dirA, LOW);  
        digitalWrite (dirB, LOW);  
        delay(300); // realiza el movimiento durante 300 milisegundos  
        k++;  
        break;
```

```
case 4: // Realiza un paso hacia la izquierda  
    Serial.println("izquierdaPaso1");  
    digitalWrite (dirA, HIGH);  
    digitalWrite (dirB, HIGH);  
    analogWrite (speedA, 245);  
    analogWrite (speedB, 10);  
    lcd.setCursor(7, 1);  
    lcd.print(" LEFT ");  
    delay(385); // realiza el movimiento durante 1 segundo  
    Serial.println("izquierdaPaso2");  
    digitalWrite (dirA, HIGH);  
    digitalWrite (dirB, HIGH);  
    analogWrite (speedA, 245);  
    analogWrite (speedB, 250);  
    delay(400); // realiza el movimiento durante 1 segundo  
    Serial.println("parada");  
    digitalWrite (dirA, LOW);  
    digitalWrite (dirB, LOW);  
    delay(300); // realiza el movimiento durante 300 milisegundos  
    k++;  
    break;
```

```

case 6: // Realiza un paso hacia la derecha
    Serial.println("derechaPaso1");
    digitalWrite (dirA, HIGH);
    digitalWrite (dirB, HIGH);
    analogWrite (speedA, 5);
    analogWrite (speedB, 250);
    lcd.setCursor(7, 1);
    lcd.print("RIGHT ");
    delay(400); // realiza el movimiento durante 1 segundo
    Serial.println("derechaPaso2");
    digitalWrite (dirA, HIGH);
    digitalWrite (dirB, HIGH);
    analogWrite (speedA, 245);
    analogWrite (speedB, 250);
    delay(400); // realiza el movimiento durante 1 segundo
    Serial.println("parada");
    digitalWrite (dirA, LOW);
    digitalWrite (dirB, LOW);
    delay(300); // realiza el movimiento durante 300 milisegundos
    k++;
    break;

case 8: // Realiza un paso hacia atras
    lcd.setCursor(7, 1);
    lcd.print(" DOWN ");
    Serial.println("atrasPaso1");
    digitalWrite (dirA, HIGH);
    digitalWrite (dirB, HIGH);
    analogWrite (speedA, 5);
    analogWrite (speedB, 250);
    delay(755); // realiza el movimiento durante 1 segundo
    Serial.println("atrasPaso2");
    digitalWrite (dirA, HIGH);
    digitalWrite (dirB, HIGH);
    analogWrite (speedA, 245);
    analogWrite (speedB, 250);
    delay(400); // realiza el movimiento durante 1 segundo
    Serial.println("parada");
    digitalWrite (dirA, LOW);
    digitalWrite (dirB, LOW);
    delay(300); // realiza el movimiento durante 300 milisegundos
    k++;
    break;

}

} while (vector[k]!=0);

Serial.println("FINISH"); //para los motores
digitalWrite (dirA, LOW);
digitalWrite (dirB, LOW);
analogWrite (speedA, 0);
analogWrite (speedB, 0);
lcd.setCursor(7, 1);
lcd.print("FINISH");
    delay(100); // realiza el movimiento durante 1 segundo
digitalWrite(LED2, HIGH); // Encendemos el LED1 para acabar el movimiento
delay(1000);
digitalWrite(LED2, LOW);
    lcd.setCursor(0, 1);
    lcd.print(" ");
break;

```

```

case 7:
    Serial.println("BORRADO");
    lcd.setCursor(7, 1);
    lcd.print("CLEAR ");
    for (int j=0; j<9; j++) { vector[j]=0; } //Llenamos de ceros el vector
    digitalWrite(LED1, HIGH); // Encendemos los LEDs e iniciar el borrado
    digitalWrite(LED2, HIGH);
    delay(500);
    digitalWrite(LED1, LOW);
    digitalWrite(LED2, LOW);
    i=0;
    lcd.setCursor(6, 1);
    lcd.print(" ");
    lcd.setCursor(0, 2);
    lcd.print(" ");
    lcd.setCursor(0,3);
    lcd.print("  ARRAY CLEAR  ");
    delay(1000);
    lcd.setCursor(0, 3);
    lcd.print(" ");
    break;

```

```

case 9:
    lcd.setCursor(0, 2);
    lcd.print("Array:");
    lcd.setCursor(6, 2);
    for (int h=0;h<9;h++){
        if (vector[h]==2) {    lcd.print("U,"); }
        if (vector[h]==4) {    lcd.print("L,"); }
        if (vector[h]==6) {    lcd.print("R,"); }
        if (vector[h]==8) {    lcd.print("D,"); }
    } // Sacar vector en pantalla
    break;

```

```

    }
}
}

```