

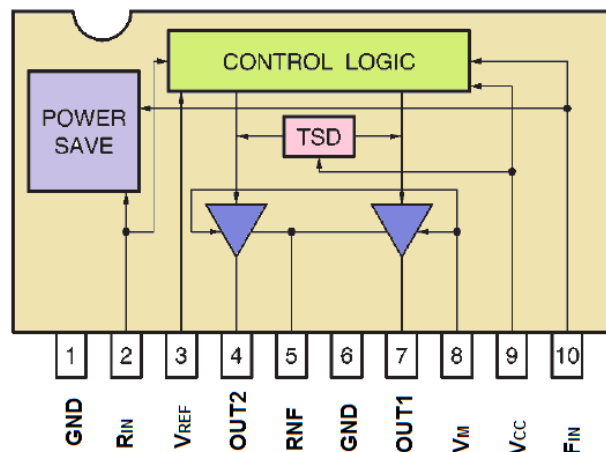
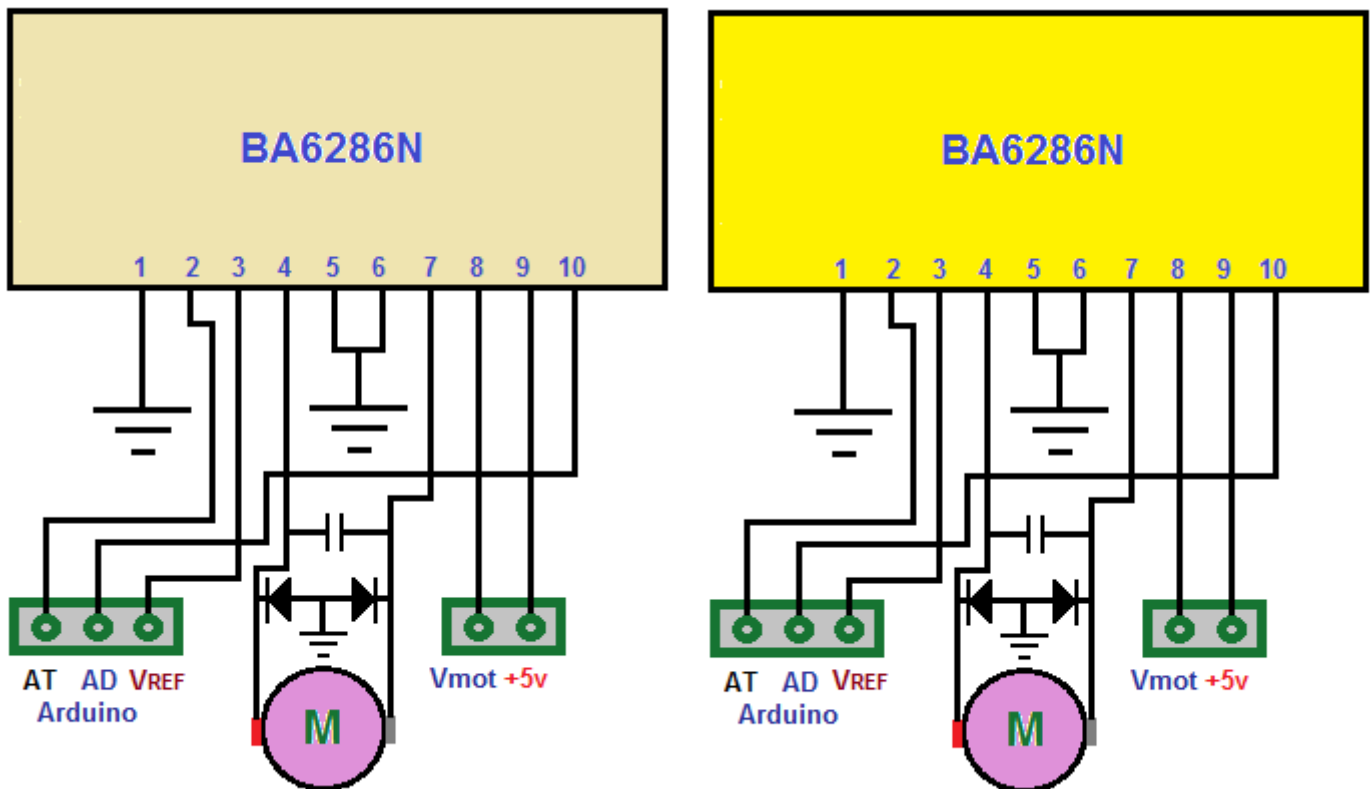
**Descripción**

El proyecto realiza la gestión de la tracción con una placa Arduino Mega 2560 encima de un chasis oruga tipo Rover. Es el primero de una serie de módulos que se irán añadiendo a dicho Rover.

El procedimiento de la gestión de la tracción es el siguiente:

- Establecemos los pines que vamos a utilizar para la gestión del movimiento
- Configuramos los pines en modo "Salida Digital".
- Realizamos los procedimientos que realizan el movimiento
- Llamamos a los procedimientos con un tiempo determinado en milisegundos a título de ejemplo.

**Esquema**



Esquema en bloque del BA6286N

## Programa

```
1  /*
2  ArduRover_Traccion
3  Este codigo controla la tracción de un Rover.
4
5  Emplea las salidas digitales 31,33,35 y 37
6
7  Este ejemplo es de dominio publico.
8  */
9
10 // Establece las constantes a utilizar
11 const int pinMotorDR=31; // Establece el pin 37 de Retroceso para el motor Derecho
12 const int pinMotorDA=33; // Establece el pin 35 de Avance para el motor Derecho
13 const int pinMotorIR=35; // Establece el pin 33 de Retroceso para el motor Izquierdo
14 const int pinMotorIA=37; // Establece el pin 31 de Avance para el motor Izquierdo
15
16 void setup()
17 {
18     pinMode(pinMotorIA, OUTPUT); //Configura pinMotorIA como salida
19     pinMode(pinMotorIR, OUTPUT); //Configura pinMotorIR como salida
20     pinMode(pinMotorDA, OUTPUT); //Configura pinMotorDA como salida
21     pinMode(pinMotorDR, OUTPUT); //Configura pinMotorDR como salida
22 }
23
24 void loop()
25 {
26     parado(2000);
27     marchaAdelante(3000);
28     giroDerecha(2000);
29     marchaAtras(3000);
30     giroIzquierda(2000);
31     marchaAdelante(1000);
32     giroIzquierda(2500);
33     giroDerecha(1900);
34     marchaAtras(1000);
35 }
36
37 void parado(int pasos)
38 {
39     digitalWrite(pinMotorIA,LOW);
40     digitalWrite(pinMotorIR,LOW);
41     digitalWrite(pinMotorDA,LOW);
42     digitalWrite(pinMotorDR,LOW);
43     delay(pasos); // Espera 1 milisegundo por cada paso
44 }
45
46 void giroDerecha(int pasos)
47 {
48     digitalWrite(pinMotorIR,HIGH);
49     digitalWrite(pinMotorDA,HIGH);
50     digitalWrite(pinMotorIA,LOW);
51     digitalWrite(pinMotorDR,LOW);
52
53     delay(pasos); // Espera 1 milisegundo por cada paso
54 }
55
56 void giroIzquierda(int pasos)
57 {
58     digitalWrite(pinMotorIA,HIGH);
59     digitalWrite(pinMotorDR,HIGH);
60     digitalWrite(pinMotorIR,LOW);
61     digitalWrite(pinMotorDA,LOW);
62     delay(pasos); // Espera 1 milisegundo por cada paso
63 }
64
65 void marchaAdelante(int pasos)
66 {
67     digitalWrite(pinMotorIA,HIGH);
68     digitalWrite(pinMotorDA,HIGH);
69     digitalWrite(pinMotorIR,LOW);
70     digitalWrite(pinMotorDR,LOW);
71     delay(pasos); // Espera 1 milisegundo por cada paso
72 }
73
74 void marchaAtras(int pasos)
75 {
76     digitalWrite(pinMotorIR,HIGH);
77     digitalWrite(pinMotorDR,HIGH);
78     digitalWrite(pinMotorIA,LOW);
79     digitalWrite(pinMotorDA,LOW);
80     delay(pasos); // Espera 1 milisegundo por cada paso
81 }
```

## Fotos

