Tanque(Arial,22,N)

Presentado por:
Lucía Pelegrín Moraz

Tutor:
Nombre Apellido1 Apellido2

Ciclo de Grado Superior
Mantenimiento Electrónico
Grupo: 2ME
Curso: 2024/2025

Índice:

1. Pliego de condiciones:

Se va a realizar un vehículo tipo tanque que cumpla con las siguientes condiciones:

- Diseño del vehículo.
- Funcionamiento a dos motores.
- Control a base de un microcontrolador.
- Movimiento y dirección gobernados por un control remoto.

2. Descripción del sistema:

Justificación del pliego de condiciones:

Se han realizado dos programas diferentes para el control de forma remota, a través de la comunicación entre dos módulos de radiofrecuencia 433Mhz. HC12, para el control de los dos motores del tanque por dos joysticks PS2, de forma que cada joystick controle uno de los motores de forma independiente, para un control más eficiente.
El programa de emisión recoge los datos de cada joystick y los envía separados por un "_". Posteriormente, el programa de recepción recoge, separa y adecúa los datos para utilizarlos para gobernar los motores del tanque.

Cálculos y justificación de cálculos:

Este proyecto se divide en dos componentes principales: el mando y el vehículo, los cuales trabajan de manera conjunta para garantizar su funcionamiento. En el caso del mando, se utilizan dos joysticks PS2 que trabajan con tres valores: el movimiento en el eje X, el movimiento en el eje Y y la pulsación del interruptor. Sin embargo, solo se emplea el valor correspondiente al eje X, por lo que los pines asociados a los valores restantes no se conectan a la placa Arduino. De esta manera, los joysticks se utilizan como potenciómetros, pero con la ventaja de que, al ser joysticks y no potenciómetros tradicionales, regresan automáticamente a una posición de reposo que proporciona un valor intermedio. Este valor se interpreta como posición de stop para los motores. El mando está controlado por una placa Arduino UNO, que se encarga de recoger los valores de los joysticks, procesarlos y estructurarlos en un mensaje. Este mensaje se genera utilizando un carácter separador "_" para diferenciar los valores de ambos joysticks. Una vez estructurado, se transmite mediante un módulo HC12 que emplea radiofrecuencia para enviarlo al vehículo, específicamente al módulo HC12 del tanque. Es importante destacar que ambos módulos HC12 deben configurarse previamente para asegurar una comunicación efectiva, y esta configuración, incluida en el código de ambos Arduinos, solo es necesaria una vez, por lo que puede eliminarse después de realizar el procedimiento inicial.

En el caso del vehículo, este está compuesto por dos placas principales: una Arduino UNO y una controladora de motores LN298. La placa Arduino recibe el mensaje enviado por el mando a través del módulo HC12 y lo procesa para controlar el funcionamiento de los motores. La estructura del mensaje, diseñada para ser interpretada por el Arduino del vehículo, utiliza el carácter separador "_" para diferenciar los valores de ambos joysticks, lo cual permite que cada motor sea controlado de forma independiente. Dado que la cantidad de caracteres recibidos es variable, se implementa un algoritmo que emplea una variable booleana para identificar el carácter separador. Este algoritmo funciona determinando que los valores recibidos antes del carácter "_" pertenecen al primer joystick, mientras que los valores posteriores corresponden al segundo

joystick. El carácter separador no se almacena en las variables finales, asegurando que los datos procesados sean limpios y utilizables para el control de los motores.

El control de los motores se lleva a cabo en dos aspectos principales: la dirección y la velocidad de giro. Para determinar la dirección, se establecen condiciones en las que si el valor recibido es superior a 550, el motor gira en un sentido, mientras que si es inferior a 500, gira en el sentido contrario. Un margen entre 500 y 550 se considera como punto muerto, deteniendo el motor. En cuanto al control de velocidad, los valores recibidos en el rango de 0 a 1023, incluso dividido en dos subrangos correspondientes a cada dirección de giro, superan claramente el rango admitido de un motor, siendo el máximo 220. A causa de esto los valores recibidos de ambos joystick se mapean a un rango adecuado para los motores, que debe ser de -255 a 255. Es importante aclarar que el rango admitido de un motor es de 0 a 220 y esto no incluye valores en negativo, por lo que posteriormente, los valores se convierten a absolutos. De esta manera el rango pasa a ser de 0 a 1024 a 220, que va menguando hasta alcanzar el 0, y de nuevo incrementa hasta alcanzar 220, generando así dos escalas: una de 0 a 255 para un sentido de giro, que correspondería a los valores originales de 0 a aproximadamente 500, y otra de 0 a 255 para el sentido contrario, que correspondería a los valores originales de aproximadamente 550 a 1024. Este mapeo asegura que el punto muerto de nuestro joystick siempre corresponda a un número alrededor del 0 y ambos extremos del joystick correspondan a una velocidad máxima de 255 en ambos sentidos. Los valores procesados se emplean en condiciones específicas que determinan tanto la dirección como la velocidad de los motores, asegurando así un control preciso del movimiento del vehículo.

La placa Arduino será atornillada al chasis del tanque y conectada a la placa LN298 mediante jumpers de forma que tanto los enables como los pines de control de giro de los motores están conectados a los pines correspondientes especificados en el código. De esta manera el Arduino le da órdenes a la placa LN298 sobre cuándo y en qué dirección y velocidad debe poner en marcha los motores. Ambos enables también forman parte del código de modo que la placa LN298 solo alimente los motores cuando sea necesario.

Como añadido, se intentó realizar el mando con un ESP12s en lugar de un Arduino uno en un intento de economizar espacio, tras diseñar la placa para poder conectar todos los elementos y comprobar que el tamaño de la placa realizada era mayor a la del Arduino UNO que se pretendía sustituir y tras una breve comprobación en la datasheet de el ESP12s que especifica que la cantidad de pines analógicos del mismo es de 1, cuando es necesario minimo 2 para no perder el control de la velocidad sobre los motores, se determinó que no era rentable en espacio, materiales o por la propia pérdida de control sobre el funcionamiento del vehículo. Esta idea ha sido descartada y no completada por estas razones.

Instrucciones de funcionamiento:

Primero debemos comprobar que el cableado de ambas partes, mando y tanque, estén debidamente conectadas siguiendo los pines marcados en cada uno de los códigos presentados. Una vez comprobado que todo esté debidamente conectado, nos aseguramos de que ambos circuitos están alimentados.

Una vez hechas las comprobaciones solo queda encender el mando y el tanque, en ese orden, de los interruptores situados en ambos circuitos. Si todo está debidamente conectado se encenderá una luz en cada una de los Arduinos pertenecientes al tanque y el mando.
Una vez todo encendido notaremos que tarda unos segundos en cargar el programa, una vez el vehículo esté listo para funcionar ambos motores dan un empujón, girando simultáneamente en la misma dirección durante una fracción de giro antes de detenerse, indicando que está listo para funcionar. Sólo queda controlar su funcionamiento con los joysticks. Cada joystick debe controlar una rueda en sentido y velocidad de giro, lo que deja una completa libertad de movimiento.

Cuando deseemos apagar el tanque, deberemos apagar primero el interruptor del tanque y posteriormente el mando para cortar la alimentación antes de retirar los cables.

Software (código fuente):

Programación HC12

```
pinMode(PRG, OUTPUT);
  digitalWrite(PRG, LOW);

  delay(500);
  HC12.write("AT+B19200");
  delay(500);
  HC12.write("AT+C063");
  delay(500);
  HC12.write("AT+P4");
  delay(500);

  digitalWrite(PRG, HIGH);
```

Este código debe incluirse en ambos códigos (emisión y recepción) situado en elsetup entre la designación “GND” y el “Serialwrite(“ ”). Esto solo se debe hacer la primera vez que se carque el programa. Una vez hecho puede eliminarse esta parte del código.

Programación emisor

```
#include <SoftwareSerial.h>

#define VCC1 13
#define VCC2 12
#define GND 11
#define RX 10
#define TX 9
#define PRG 8

const int M1 = A1;
int Val_M1;

const int M2 = A0;
int Val_M2;

SoftwareSerial HC12(TX, RX);

void setup() {

  pinMode(M1, INPUT);
  pinMode(M2, INPUT);

  Serial.begin(9600);
  HC12.begin(9600);

  pinMode(VCC1, OUTPUT);
  digitalWrite(VCC1, HIGH);

  pinMode(VCC2, OUTPUT);
```

```
  digitalWrite(VCC2, HIGH);

  pinMode(GND, OUTPUT);
  digitalWrite(GND, LOW);

  pinMode(PRG, OUTPUT);
  digitalWrite(PRG, LOW);

  delay(500);
  HC12.write("AT+B9600");
  delay(500);
  HC12.write("AT+C001");
  delay(500);
  HC12.write("AT+P4");
  delay(500);
  HC12.write("AT+FU1");
  delay(500);

  digitalWrite(PRG, HIGH);

  Serial.println("emisor");
  delay(3000);
}

void loop() {

  Val_M1 = analogRead(M1);
  Val_M2 = analogRead(M2);
  Serial.print(Val_M1);
  Serial.print(" / ");
  Serial.println(Val_M2);

  HC12.print(Val_M1);
  HC12.print("_");
  HC12.println(Val_M2);
  HC12.print("");
  delay(200);

}
```

Programación receptor

```
#include <SoftwareSerial.h>

#define VCC 6
#define GND 5
#define RX  4
#define TX  3
#define PRG 2
```

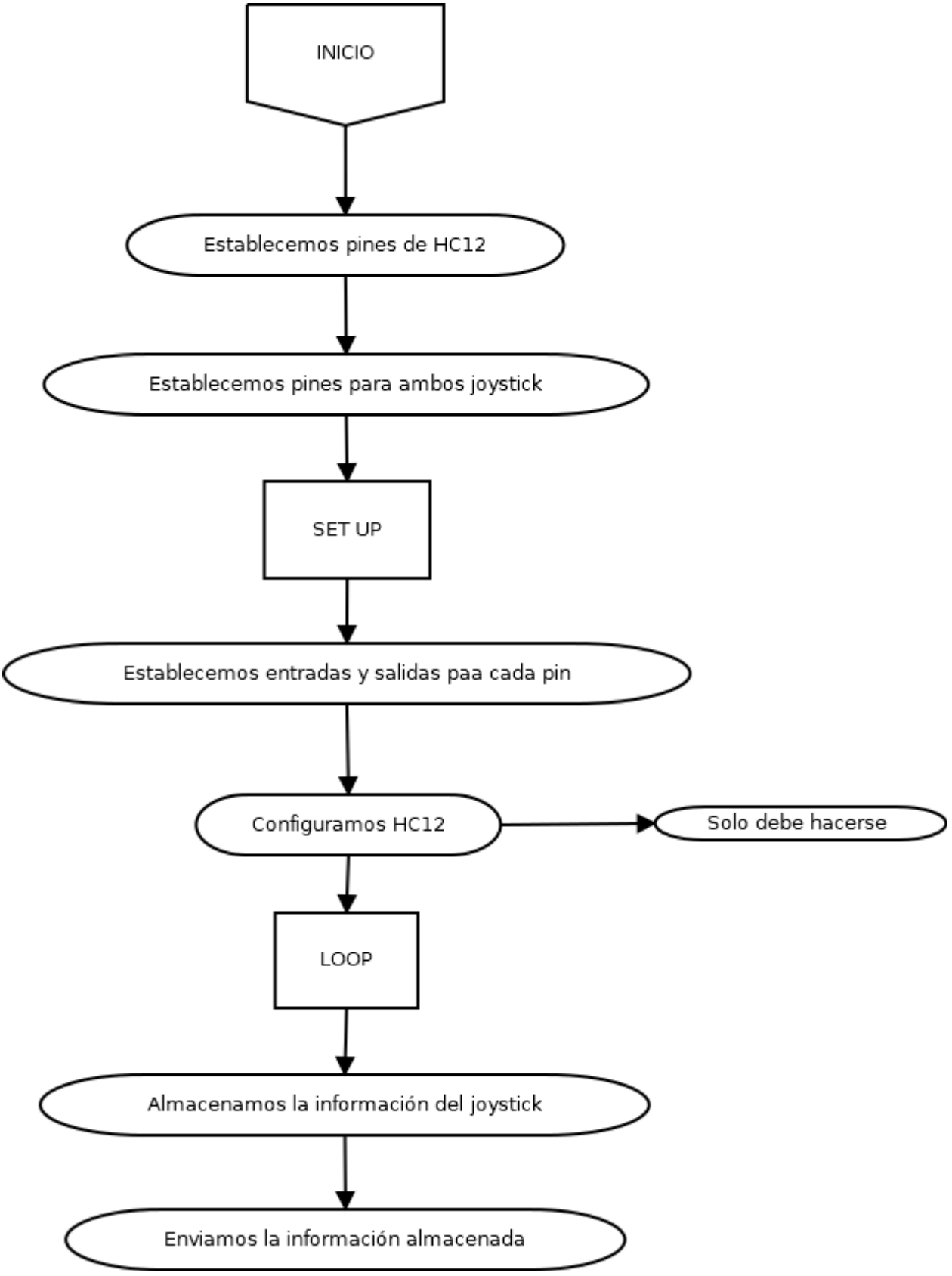
```
const int M1D=A0, M1I=A1, EN1=12; //pin motor 1
const int M2D=A4, M2I=A5, EN2=11; //pin motor 2

SoftwareSerial HC12(TX, RX); // HC-12 TX Pin, HC-12 RX Pin

  String mensaje = "";
  String mensajeA = "";
  String mensajeB = "";
  bool variable=false;
  int Val_1;
  int Val_2;
  int Valmap_1;
  int Valmap_2;
  int Valabs_1;
  int Valabs_2;

void setup()
{
  pinMode(M1D, OUTPUT);
  digitalWrite(M1D, LOW);
  pinMode(M1I, OUTPUT);
  digitalWrite(M1I, LOW);

  Serial.begin(9600);
  HC12.begin(9600);

  pinMode(VCC, OUTPUT);
  digitalWrite(VCC, HIGH);

  pinMode(GND, OUTPUT);
  digitalWrite(GND, LOW);

  pinMode(PRG, OUTPUT);
  digitalWrite(PRG, LOW);

  delay(500);
  HC12.write("AT+B9600");
  delay(500);
  HC12.write("AT+C001");
  delay(500);
  HC12.write("AT+P4");
  delay(500);
  HC12.write("AT+FU1");
  delay(500);

  digitalWrite(PRG, HIGH);

  Serial.println("receptor");
  delay(2000);
}


void loop() {
```

```
  HC12.flush();
  while (HC12.available()){
    char c = HC12.read();
    mensaje += c;
    delay(10);
  }


  mensaje.trim(); //elimina los espacios en blanco al principio y al final del mensaje
  int longitud = mensaje.length();

  //Serial.println(mensaje);

  if (longitud > 0 && longitud < 10) {
    boolean variable = false;  // Indica si ya se encontró el separador "_"
    mensajeA = "";  // Limpiamos las variables antes de procesar
    mensajeB = "";

    for (int i = 0; i < longitud; i++) {
        String separador = mensaje.substring(i, i + 1);  // Obtenemos el carácter actual

        // Si encontramos el separador "_", cambiamos el estado de variable
        if (separador == "_") {
            variable = true;
        } else if (!variable) {
            // Antes del primer "_", los caracteres se agregan a mensajeA
            mensajeA += separador;
        } else {
            // Después del primer "_", los caracteres se agregan a mensajeB
            mensajeB += separador;
        }
    }

    // Imprimimos los resultados para verificar la separación
    Serial.println("Mensaje: " + mensaje);  // Mensaje completo original
    /**
    Serial.println("MensajeA: " + mensajeA);  // Contenido antes del "_"
    Serial.println("MensajeB: " + mensajeB);  // Contenido después del "_"
    Serial.println("Longitud: " + String(longitud));  // Longitud del mensaje
    **/
}

    if(longitud>0){
      Val_1 = mensajeA.toInt();
      Val_2 = mensajeB.toInt();
      Valmap_1 = map(Val_1, 0, 1023, -255, 255);
      Valmap_2 = map(Val_2, 0, 1023, -255, 255);
      Valabs_1 = abs(Valmap_1);
      Valabs_2 = abs(Valmap_2);
    }
    //Serial.println(Val_1 + " _ " + Val_2);

    //control motor 1
      if(Val_1>(550)){
```

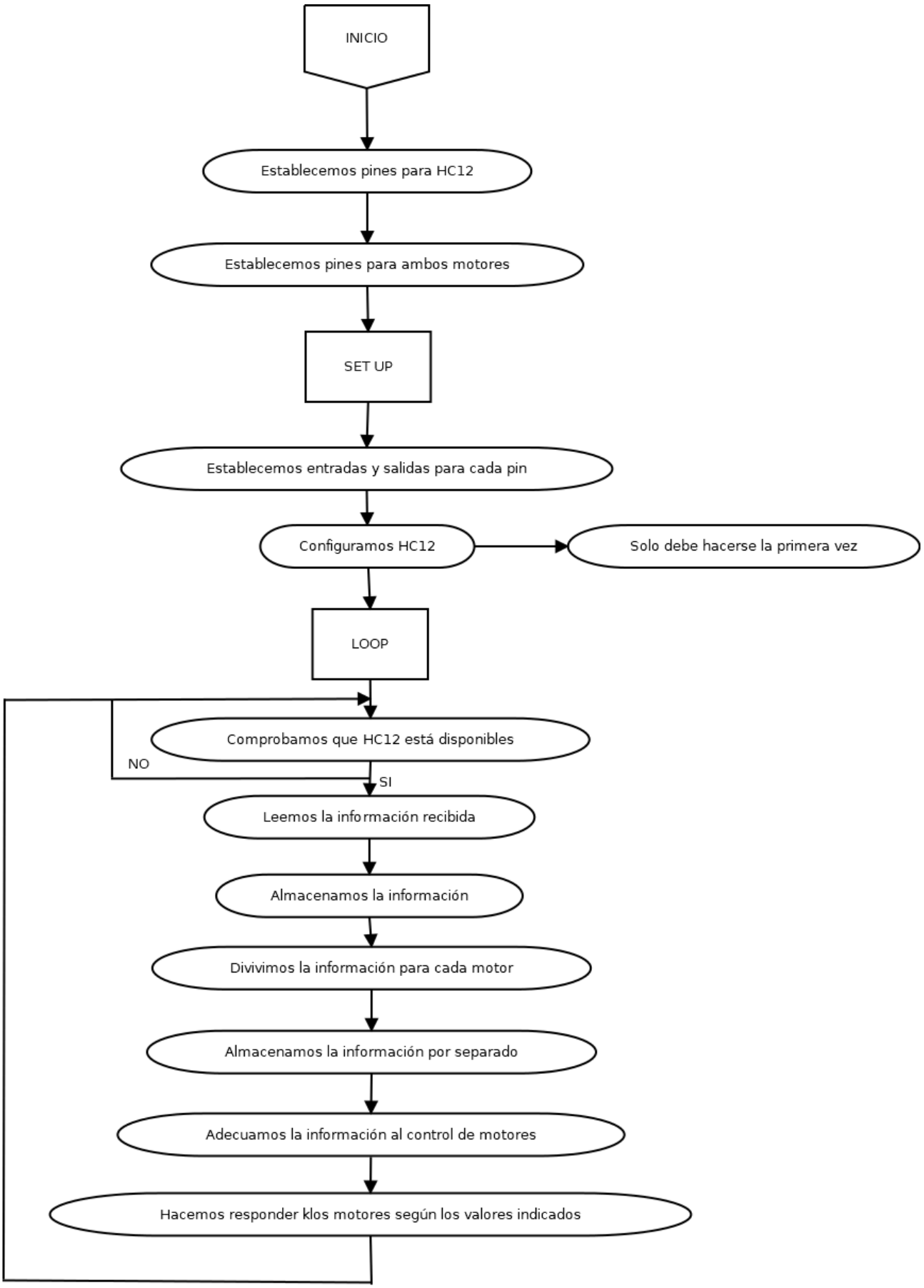
```
      digitalWrite(EN1, HIGH);
      analogWrite(M1D, Valabs_1);
      //Serial.println("M1 DERECHA");
    } else if(Val_1<550 && Val_1>500){
      digitalWrite(M1D, LOW);
      analogWrite(M1I, LOW);
      //Serial.println("STOP M1");
    } else{
      digitalWrite(EN1, HIGH);
      analogWrite(M1I, Valabs_1);
      //Serial.println("M1 IZQUIERDA");
    }

  //Control M2
    if(Val_2>(550)){
      digitalWrite(EN2, HIGH);
      analogWrite(M2D, Valabs_2);
      //Serial.println("M2 DERECHA");
    } else if(Val_2<550 && Val_2>500){
      digitalWrite(M2D, LOW);
      analogWrite(M2I, LOW);
      //Serial.println("STOP M2");
    } else{
      digitalWrite(EN2, HIGH);
      analogWrite(M2I, Valabs_2);
      //Serial.println("M2 IZQUIERDA");
    }

  mensaje = "";
  mensajeA = "";
  mensajeB = "";

  HC12.flush();

}
```

Funcionamiento general del programa utilizando un diagrama de flujo:

Emisor:



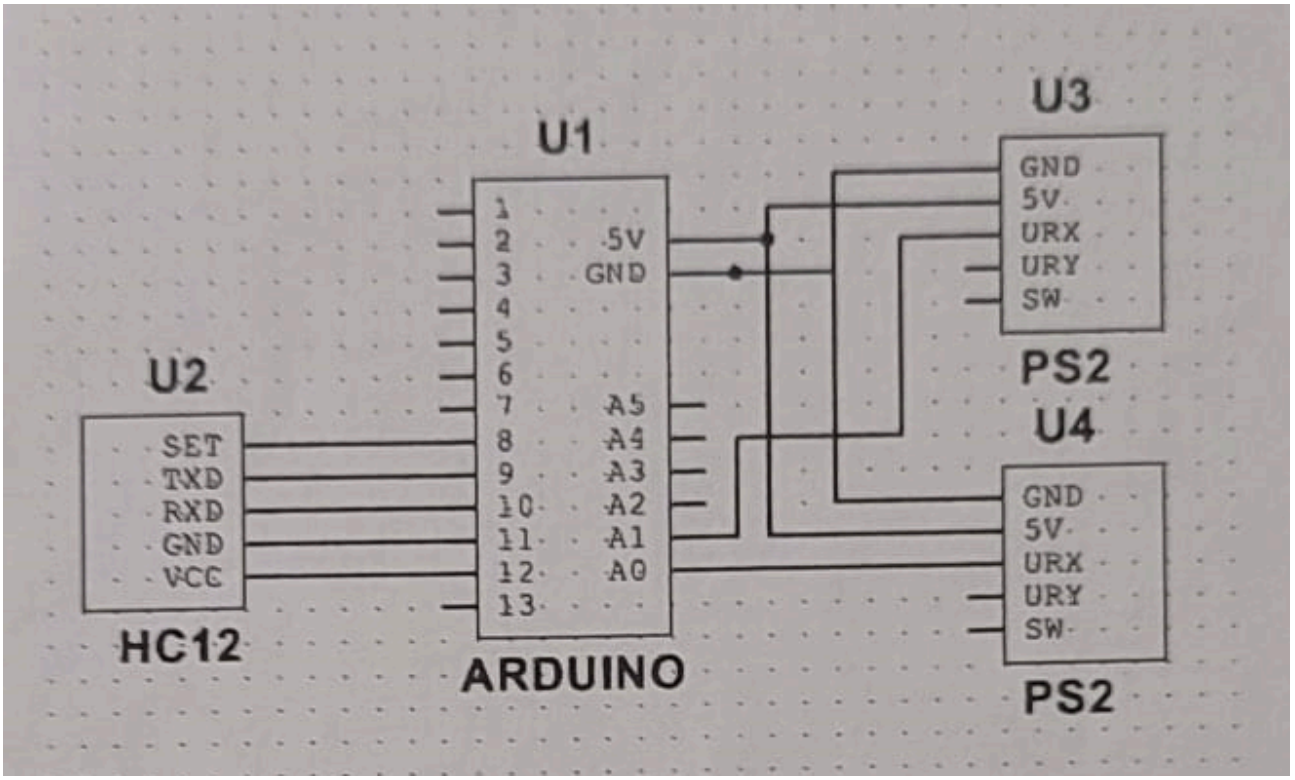
Receptor:


INICIO
Establecemos pines para HC12
Establecemos pines para ambos motores
SET UP
Establecemos entradas y salidas para cada pin
Configuramos HC12
Solo debe hacerse la primera vez
LOOP
Comprobamos que HC12 está disponibles
NO
SI
Leemos la información recibida
Almacenamos la información
Divivimos la información para cada motor
Almacenamos la información por separado
Adecuamos la información al control de motores
Hacemos responder klos motores según los valores indicados

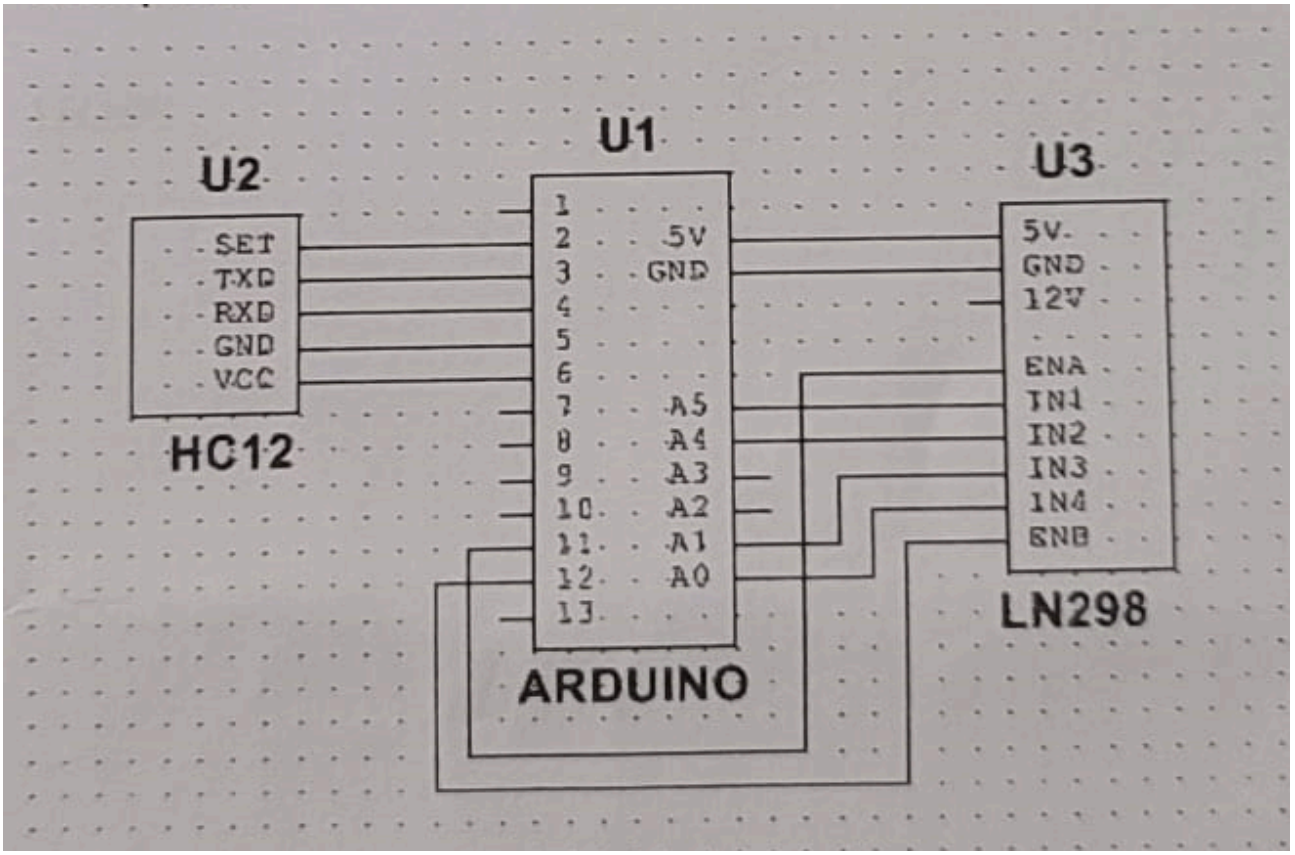
Planos

Eléctrico:

Emisor:

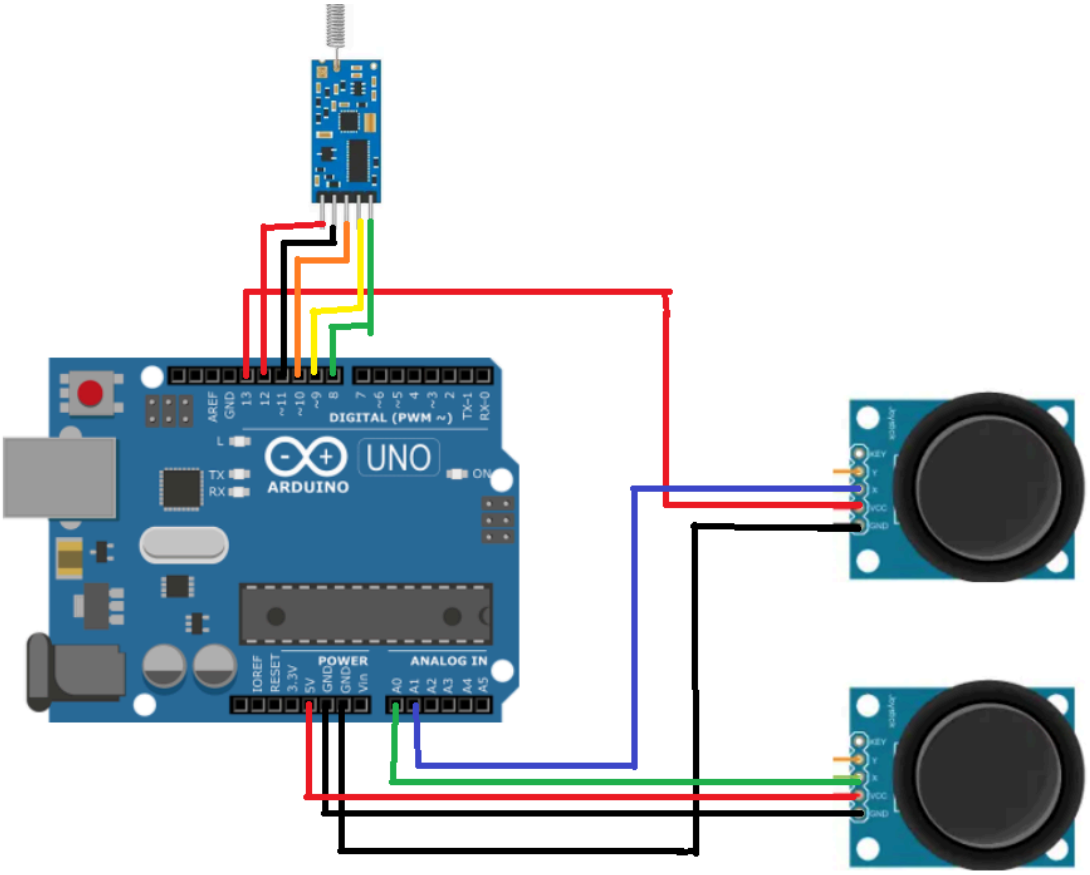


Receptor:

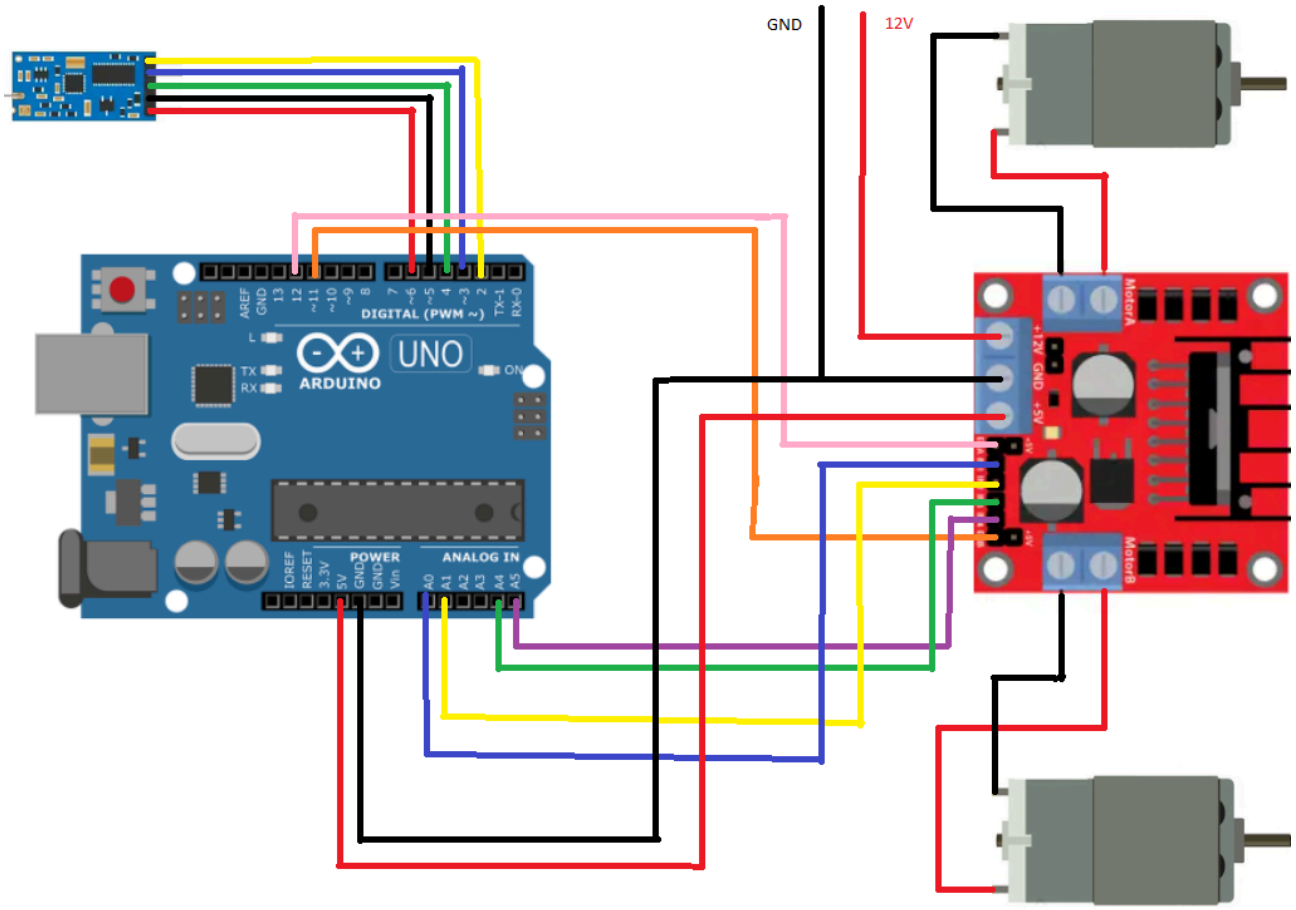

U1
U2
U3
SET
TXD
RXD
GND
VCC
HC12
1
2
3
4
5
6
7
8
9
10
11
12
13
5V
GND
A5
A4
A3
A2
A1
A0
ARDUINO
5V
GND
12V
ENA
IN1
IN2
IN3
IN4
ENB
LN298

Topográfico:

Emisor:



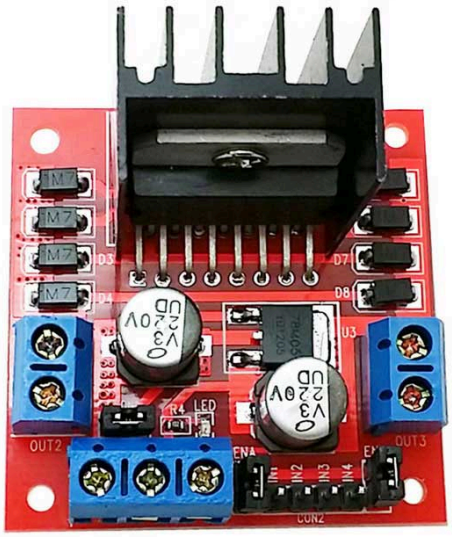
Receptor:


GND
12V
DIGITAL (PWM ~)
UNO
ARDUINO
POWER
ANALOG IN
MotorA
MotorB
+12V
GND
+5V

Placas utilizadas:

Arduino:

LN298:



Presupuesto:

Componentes:

en este presupuesto no se están contando los materiales utilizados en el desarrollo de la placa personalizada para el ESP12s.

Componente	Cantidad	Precio individual	Precio total
Placa Arduino	2	24€	48€
Cables macho-hembra	6	0.18€	4.80€
Cables macho-macho	8	0.23€	1.84€
interruptor	2	2€	4€
Joystick	2	9.29€	18.58€
HC12	2	8.44€	16.89€
Chasis tanque	1	88.38€	88.39€
Placa LN298	1	3.66€	3.66€
		TOTAL	**186.16**

Mano de obra:

Horas	Precio por hora	total
60	7€	**420€**

Precio total:

Materiales	Mano de obra	Total
186.16€	420€	**606.16€**