CATEGORÍAS DEL BLOG

Tutoriales

Noticias

ÚLTIMOS COMENTARIOS

Troubleshooter

en Configuración del módulo bluetooth HC-05...

juan carlos

en Tutorial cinta LED RGB WS2812B

X

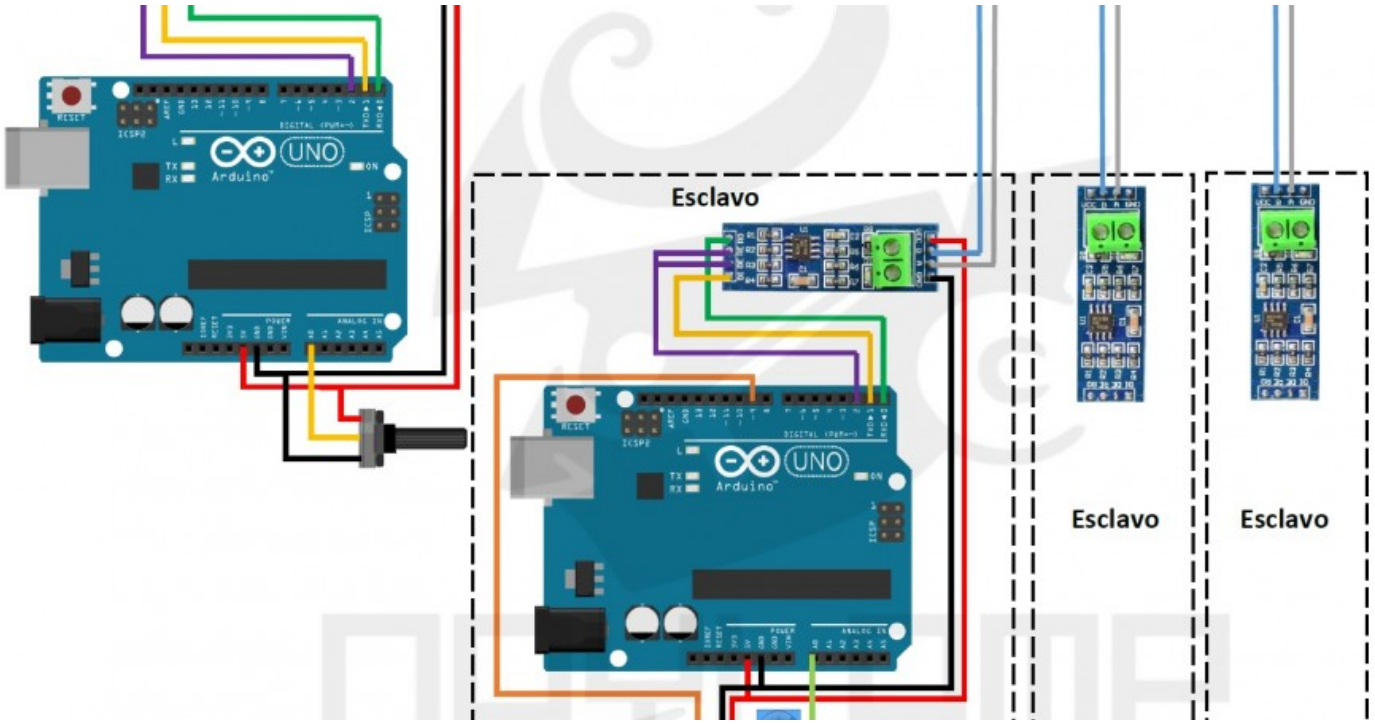
en Tutorial Módulo GPS con Arduino

Andrés F.

en Tutorial de Uso del Módulo L298N

BÚSQUEDA DE BLOGS

216562



Explicaremos como utilizar los Módulos RS485 e implementaremos varios ejemplos para comunicar 2 Arduinos a través de RS485, desarrollaremos la comunicación simplex, full duplex y half duplex.

El estándar industrial RS485 o también conocido como **EIA-485** es un estándar de capa física según el modelo OSI. Es decir, no pone normas ni restricciones sobre el contenido, forma, o codificación de los mensajes enviados. Utilizando RS485 como capa física es posible implementar un bus de campo industrial tipo MODBUS para comunicación entre equipos industriales o DMX para el control de luces entre otras aplicaciones. El estándar RS485 es ampliamente utilizado por su robustez, fácil implementación y buenas prestaciones.

En este tutorial trabajaremos el tema de Comunicación Serial a sobre RS485 con Arduino.

RS485 está definido como un sistema de comunicación en bus diferencial multipunto, es ideal para transmitir a velocidades medianas sobre largas distancias (35 Mbit/s hasta 10 metros y 100 kbit/s en 1200 metros) y a través de canales ruidosos, ya que al emplear voltajes diferenciales reduce los ruidos eléctricos que aparecen en la línea de transmisión. El medio físico de transmisión (cableado) es el cable par trenzado que admite hasta 32 estaciones en 1 solo bus, con una longitud máxima de 1200 metros operando entre 300 y 19200 bit/s bajo comunicación half-duplex (semiduplex).

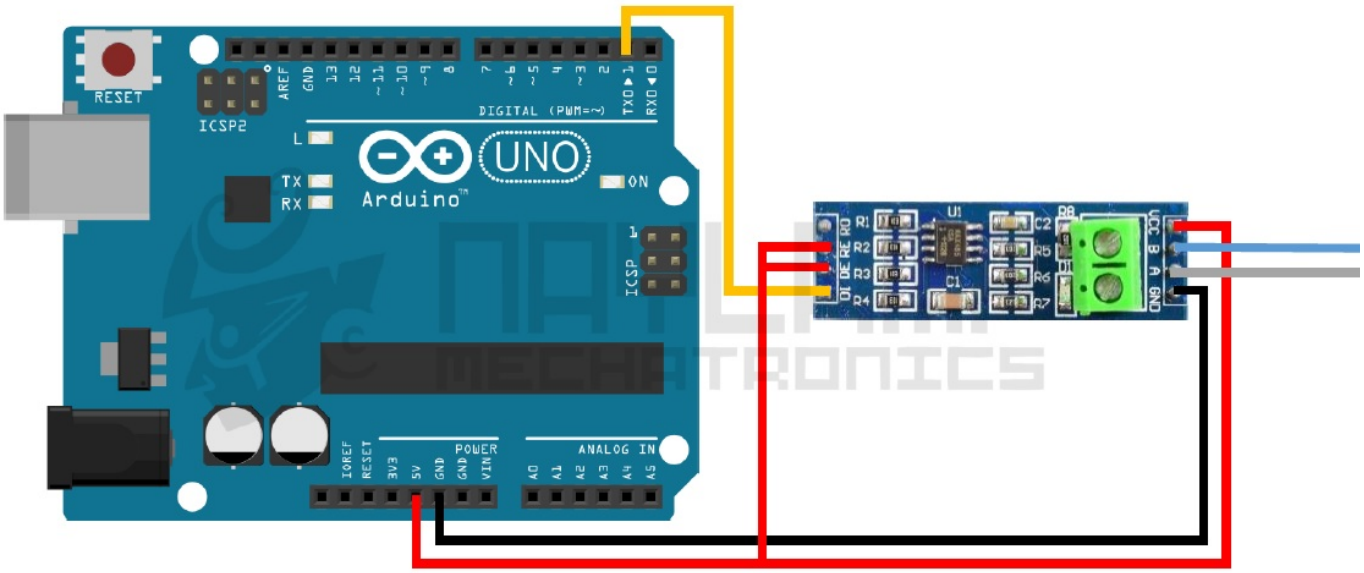
Para la implementación de los ejemplos del tutorial utilizaremos los **Módulos RS485 a Serial TTL**

El módulo tiene acceso a todos los pines del chip MAX485 y tiene todos los componentes adicionales para su correcto funcionamiento. Los pines "A" y "B" se pueden conectar también desde la bornera.

Las pines A y B son por donde se envían y reciben los datos, B es la negación en voltaje de A, se envían los mismos datos pero en la línea B están negados, de esta forma se pueden restar ambas señales y eliminar el ruido y quedarnos solo con la señal de datos. El encargado de esta transmisión diferencial es el chip MAX485.

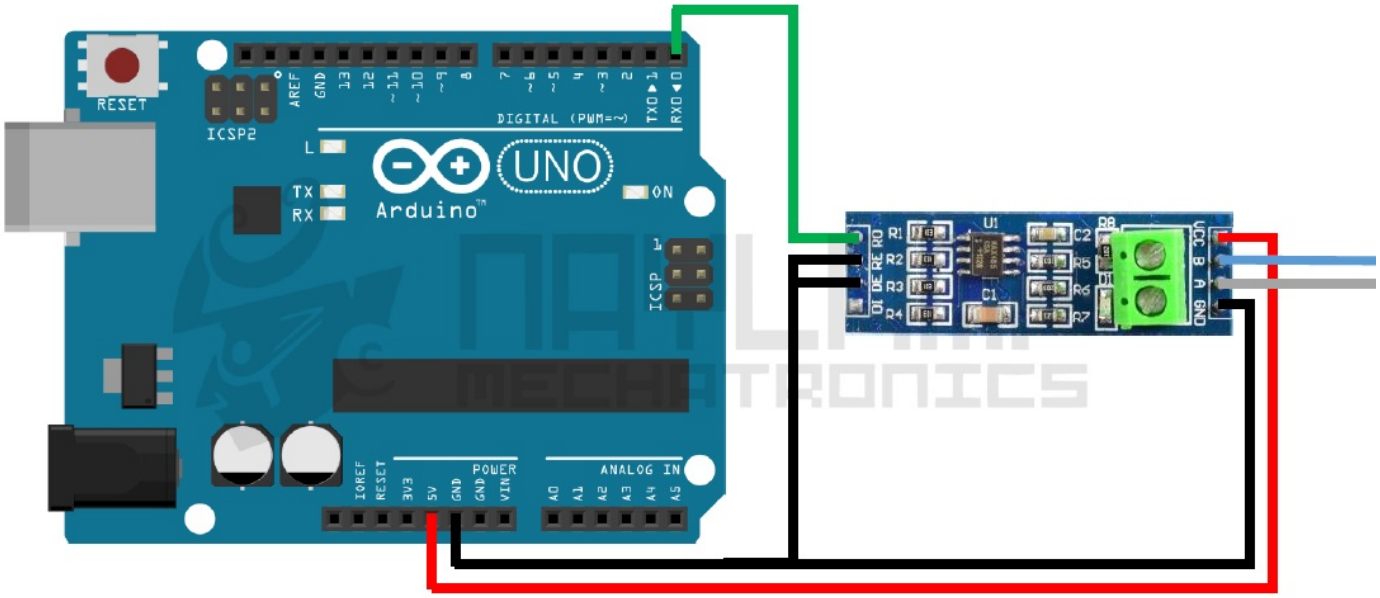
Desde el punto de vista del microcontrolador o Arduino, la comunicación se trabaja como una comunicación serial, trabajamos con los pines RX y TX; y dependiendo del tipo de conexión se puede usar un pin más para especificar si estamos enviando o recibiendo datos.

En esta configuración el modulo solo trabaja como transmisor, para que el modulo sepa que las salida A B se van a comportar como salida de datos, se tiene que conectar a 5V los pines RE y DE. Desde el Arduino se envían los datos hacia el pin DI (Data Input) del módulo y este transmitirá los datos por los pines AB



Usar el módulo RS485 como receptor

Al conectar los pines RE y DE el modulo se comporta como Receptor, y los datos recibidos por AB estarán presentes en el pin RO(Receiver Output), conectando el pin RO del módulo al RX de nuestro Arduino podemos leer los datos recibidos.

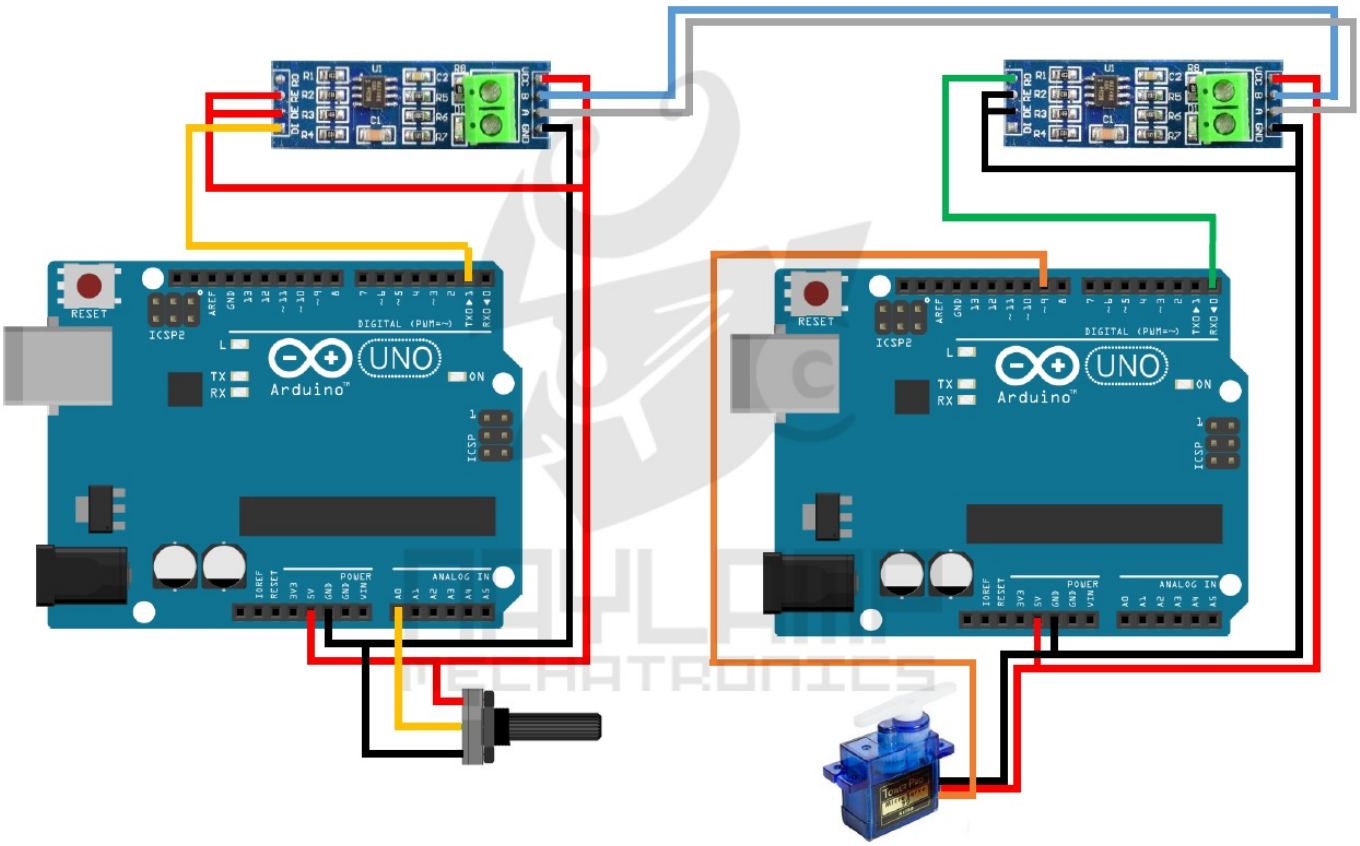


RS485

Una comunicación simplex es una comunicación unidireccional, en este caso un Arduino se comporta solo como transmisor y el otro solo como receptor, a nivel de programación es como si estuviéramos trabajando con una comunicación serial, pero en un solo sentido. Uno envía y el otro solo recibe datos.

Veamos un ejemplo:

Desde un Arduino a través de un potenciómetro moveremos un servomotor que estará conectado en otro Arduino, solo dos cables (salidas A y B del RS485) unirán a los Arduinos, si la distancia es larga se recomienda usar cable trenzado.



El código del transmisor es el siguiente:

```
  Serial.begin(9600);
}

void loop()
{
  int lectura = analogRead(0);//leemos el valor del potenciómetro (de 0 a 1023)
  byte angulo= map(lectura, 0, 1023, 0, 180);     // escalamos la lectura a un valor de ángulo
(entre 0 y 180)
  Serial.write(angulo); //enviamos el ángulo correspondiente
  delay(50);
}
```

El código del Arduino receptor es el siguiente:

```
#include <Servo.h>

Servo myservo;  // creamos el objeto servo

void setup()
{
  Serial.begin(9600);
  myservo.attach(9);  // asignamos el pin 9 para el servo.
}

void loop()
{

  if (Serial.available()) {
    int angulo = Serial.read(); //Leemos el dato recibido
    if(angulo<=180) //verificamos que sea un valor en el rango del servo
    {
      myservo.write(angulo); //movemos el servomotor al ángulo correspondiente.
    }
  }
}
```

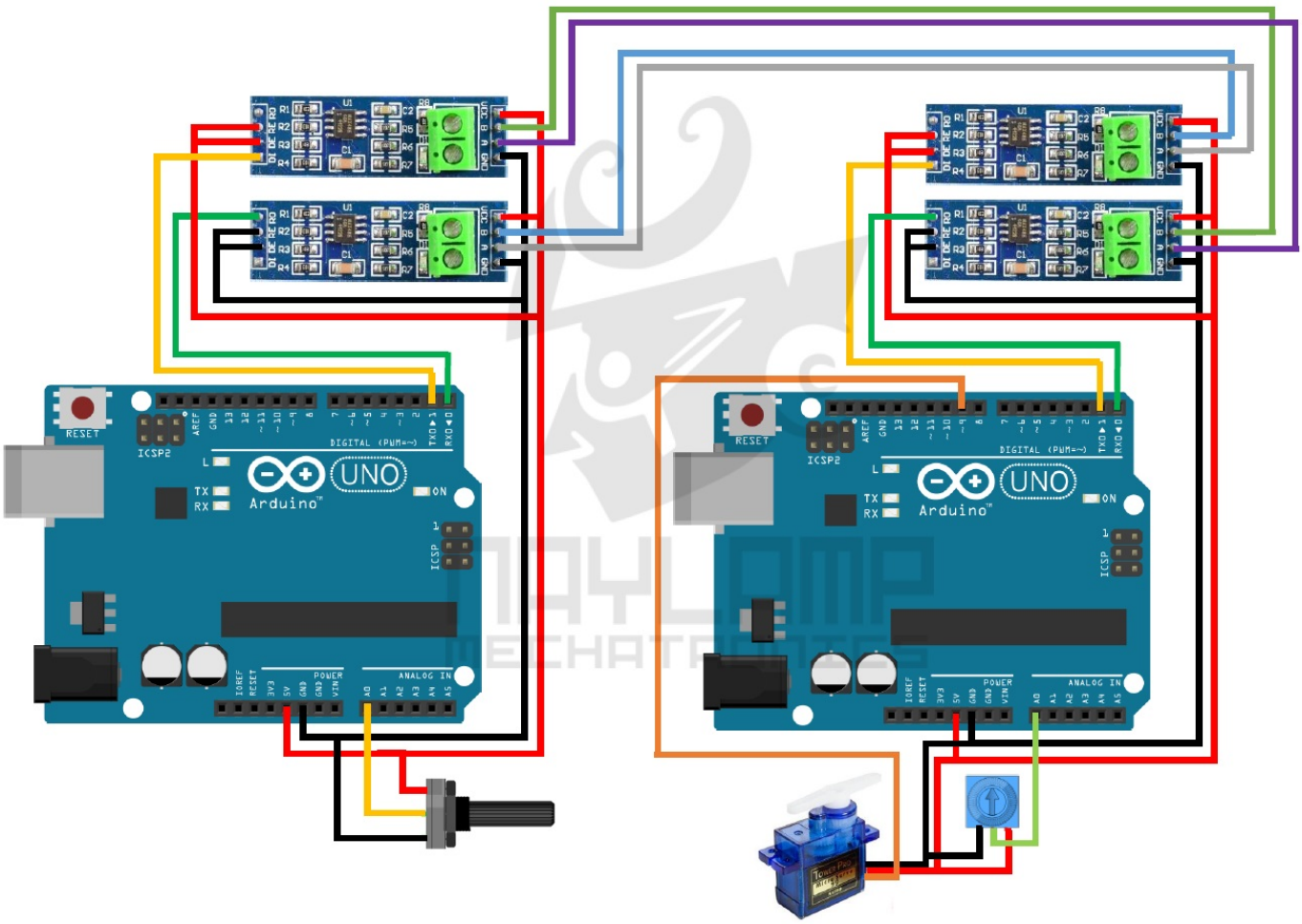
Como se observa es una simple comunicación serial, en el Arduino transmisor se hace la lectura del potenciómetro se lo escala y se lo envía serialmente, en el receptor recibimos el dato y movemos el servomotor.

De esta forma podemos realizar una comunicación entre dos Arduino pero en una sola dirección, para hacerlo en dos direcciones tenemos dos formas, half-duplex y full-duplex

En este caso necesitamos agregar otro par de líneas más, en total unirían a los Arduino 4 líneas, un par son para transmitir (TX) y otro par para recibir (RX).

En el siguiente ejemplo desde un Arduino no solo enviaremos datos para mover un servomotor sino también recibiremos datos de un sensor, usaremos un potenciómetro para simular el sensor.

Las conexiones serían las siguientes:



Al Arduino de la izquierda, lo llamaremos Maestro, pues es el Arduino principal y quien administrara todas las órdenes, mientras que el segundo Arduino lo denominaremos Esclavo; esta no es específicamente la definición de Maestro/Esclavo en una comunicación RS485 pero usaremos estas denominaciones para saber a qué Arduino nos estamos refiriendo.

El código del Arduino Maestro.

```
{
  Serial.begin(9600);
  pinMode(ledPin, OUTPUT);//inicializamos el pin del Led como salida
}

void loop()
{
  if(Serial.available())
  {
    if(Serial.read()=='i') //si recibimos el inicio de trama
    {
       int dato=Serial.parseInt(); //recibimos valor numérico
       if(Serial.read()=='f') //Si el fin de trama es el correcto
       {
         funcion(dato);  //Realizamos la acción correspondiente
       }
    }
  }

  int lectura = analogRead(0);//leemos el valor del potenciómetro (de 0 a 1023)
  int angulo= map(lectura, 0, 1023, 0, 180); // escalamos la lectura a un valor de ángulo (entre 0
y 180)
  //---enviamos el ángulo para mover el servo------
  Serial.print("I"); //inicio de trama
  Serial.print("S"); //S para indicarle que vamos a mover el servo
  Serial.print(angulo); //ángulo  o dato
  Serial.print("F"); //fin de trama
  //----------------------------
  delay(50);
  //---solicitamos una lectura del sensor----------
  Serial.print("I"); //inicio de trama
  Serial.print("L"); //L para indicarle que vamos a Leer el sensor
  Serial.print("F"); //fin de trama
  //-----------------------------------------------
  delay(50);

}

//esta función  puede variar de acuerdo a su necesidad
void funcion(int dato)
{
  if(dato>500)
  digitalWrite(ledPin, HIGH);
  else
  digitalWrite(ledPin, LOW);
}
```

Código del Arduino Esclavo:

```
Servo myservo;  // creamos el objeto servo

void setup()
{
  Serial.begin(9600);
  myservo.attach(9);  // asignamos el pin 9 para el servo.
}

void loop()
{
  if(Serial.available()>0)
  {
    if(Serial.read()=='I') //Si recibimos el inicio de trama
    {
      char funcion=Serial.read(); //leemos el carácter de función
      //---Si el carácter de función es una S entonces la trama es para mover el motor-----------
      if(funcion=='S')
       {
           int angulo=Serial.parseInt(); //recibimos el ángulo
           if(Serial.read()=='F') //Si el fin de trama es el correcto
           {
             if(angulo<=180) //verificamos que sea un valor en el rango del servo
              {
                myservo.write(angulo); //movemos el servomotor al ángulo correspondiente.
              }
           }
       }
       //---Si el carácter de función  es L entonces el maestro está solicitando una lectura del
sensor
       else if(funcion=='L')
       {
          if(Serial.read()=='F') //Si el fin de trama es el correcto
           {
             int lectura = analogRead(0); //realizamos  la lectura del sensor
             //----enviamos la respuesta-----
             Serial.print("i"); //inicio de trama
             Serial.print(lectura); //valor del sensor
             Serial.print("f"); //fin de trama
             //-----------------
           }
       }
    }
  }
  delay(10);
}
```

[Inicio de trama][Función][Valor][Fin de trama]

En nuestro caso el inicio de trama es el carácter 'A' , la función es el carácter S o L para indicar que vamos a mover el servo o solicitar una lectura del sensor, el [valor] solo estará presente cuando la función necesite enviar una dato, y el fin de trama que usamos es el carácter F. Unos ejemplos de esta trama serian: "IS90F","IS120F","ILF", etc.

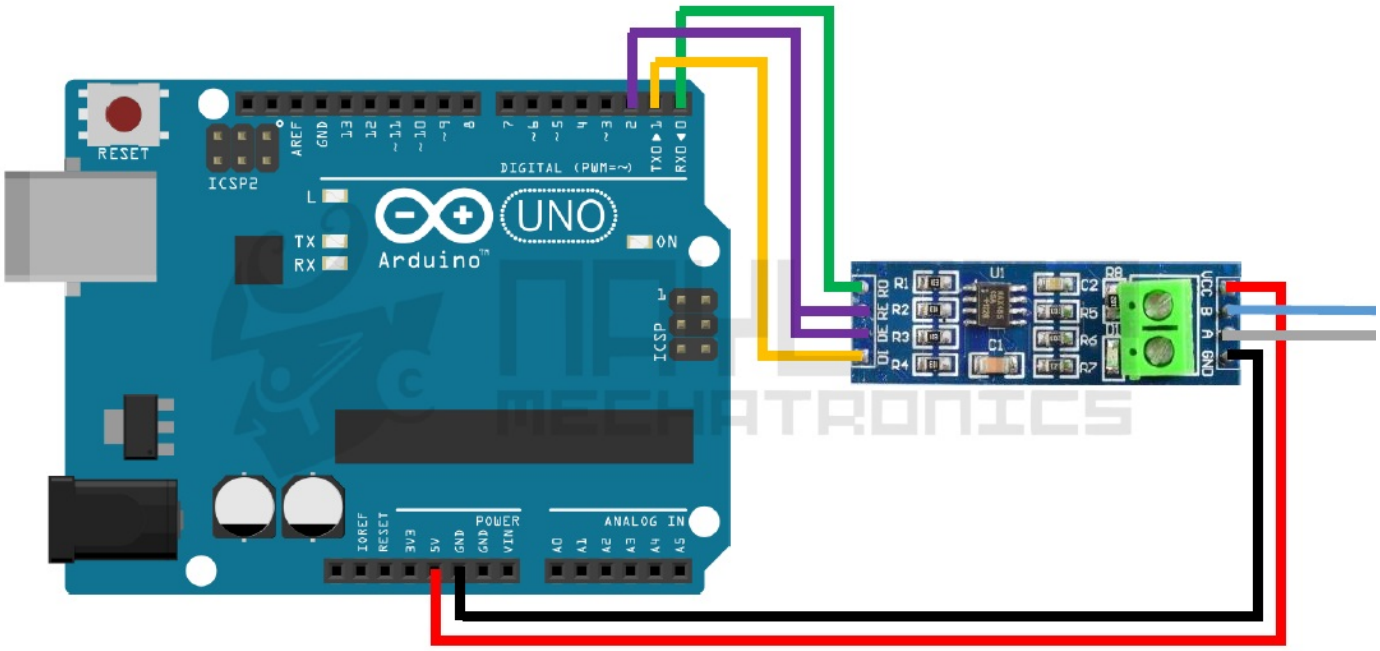
El esclavo interpreta esta trama y realiza la función correspondiente, si es una función que necesite responder, la trama de respuesta es:

[Inicio de trama][Valor][Fin de trama] , como por ejemplo "i865f", "i64f"

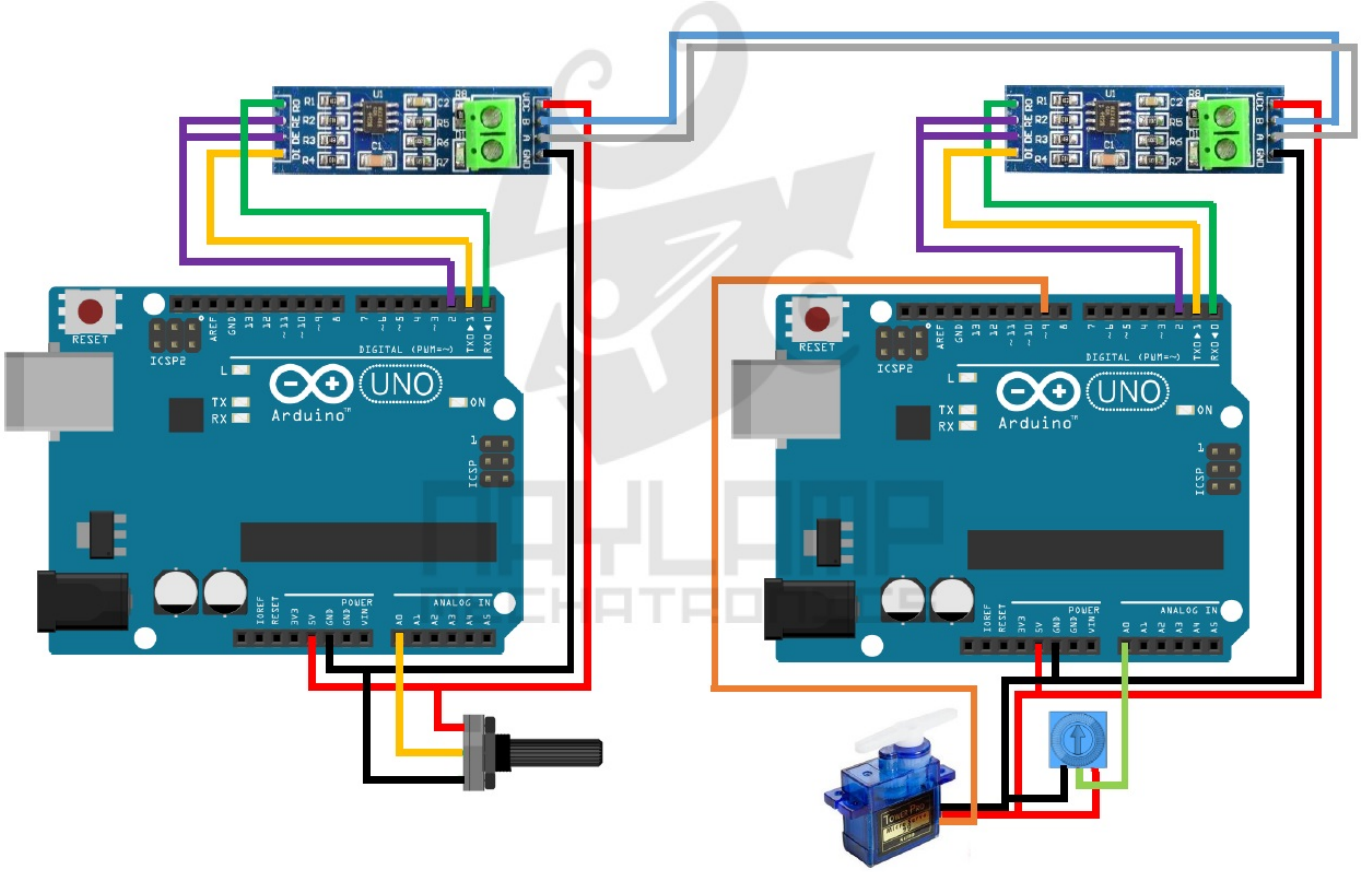
Ej 3. Comunicación half dúplex entre dos Arduinos

En una comunicación half dúplex utiliza un solo canal para comunicarse, en un momento por el canal se transmiten datos y en otro momento se reciben datos, pero nunca podremos transmitir y recibir a la vez.

Para realizar esta comunicación los pines DE y RE del módulo RS485 deben ir conectados al Arduino, con esto desde el programa podemos establecer al módulo como transmisor o receptor



El siguiente ejemplo hace lo mismo que el ejemplo anterior solo que esta vez se usa un módulo rs485 por Arduino y un par de cables para comunicarse.



Al igual que en el caso anterior el Arduino de la izquierda será el maestro y el de la derecha será el esclavo.

Código del Maestro:

```
void setup()
{
  Serial.begin(9600);
  Serial.setTimeout(100);//establecemos un tiempo de espera de 100ms
  //inicializamos los pines
  pinMode(ledPin, OUTPUT);
  pinMode(EnTxPin, OUTPUT);
  digitalWrite(ledPin, LOW);
  digitalWrite(EnTxPin, HIGH);
}

void loop()
{

  int lectura = analogRead(0);//leemos el valor del potenciómetro (de 0 a 1023)
  int angulo= map(lectura, 0, 1023, 0, 180);// escalamos la lectura a un valor de ángulo (entre 0 y
180)
  //---enviamos el ángulo para mover el servo------
  Serial.print("I"); //inicio de trama
  Serial.print("S"); //S para indicarle que vamos a mover el servo
  Serial.print(angulo); //ángulo  o dato
  Serial.print("F"); //fin de trama
  //----------------------------
  delay(50);
  //---solicitamos una lectura del sensor----------
  Serial.print("I"); //inicio de trama
  Serial.print("L"); //L para indicarle que vamos a Leer el sensor
  Serial.print("F"); //fin de trama
  Serial.flush();    //Esperamos hasta que se envíen los datos
  //----Leemos la respuesta del Esclavo-----
  digitalWrite(EnTxPin, LOW); //RS485 como receptor
  if(Serial.find("i"))//esperamos el inicio de trama
  {
      int dato=Serial.parseInt(); //recibimos valor numérico
      if(Serial.read()=='f') //Si el fin de trama es el correcto
       {
         funcion(dato);  //Realizamos la acción correspondiente
      }

  }
  digitalWrite(EnTxPin, HIGH); //RS485 como Transmisor
  //----------fin de la respuesta-----------

}
void funcion(int dato)
{
  if(dato>500)
  digitalWrite(ledPin, HIGH);
```

```
}
```

Código de Esclavo:

```
Servo myservo;  // creamos el objeto servo
const int EnTxPin =  2;
void setup()
{
  Serial.begin(9600);
  myservo.attach(9);  // asignamos el pin 9 para el servo.
  pinMode(EnTxPin, OUTPUT);
  digitalWrite(EnTxPin, LOW); //RS485 como receptor
}

void loop()
{
  if(Serial.available())
  {
    if(Serial.read()=='I') //Si recibimos el inicio de trama
    {

      char funcion=Serial.read();//leemos el carácter de función
      //---Si el carácter de función es una S entonces la trama es para mover el motor-----------
      if(funcion=='S')
       {
           int angulo=Serial.parseInt(); //recibimos el ángulo
           if(Serial.read()=='F') //Si el fin de trama es el correcto
           {
             if(angulo<=180) //verificamos que sea un valor en el rango del servo
              {
                myservo.write(angulo); //movemos el servomotor al ángulo correspondiente.
              }
           }
       }
       //---Si el carácter de función  es L entonces el maestro está solicitando una lectura del
sensor---
       else if(funcion=='L')
       {
          if(Serial.read()=='F') //Si el fin de trama es el correcto
           {
             int lectura = analogRead(0);  //realizamos  la lectura del sensor
             digitalWrite(EnTxPin, HIGH);  //rs485 como transmisor
             Serial.print("i"); //inicio de trama
             Serial.print(lectura); //valor del sensor
             Serial.print("f"); //fin de trama
             Serial.flush(); //Esperamos hasta que se envíen los datos
             digitalWrite(EnTxPin, LOW); //RS485 como receptor
           }
       }
    }
  }
  delay(10);
```

Como se observa el código cada vez que vamos a escribir o leer datos activamos o desactivamos respectivamente el pin que va conectado a DE y RE del módulo RS485.

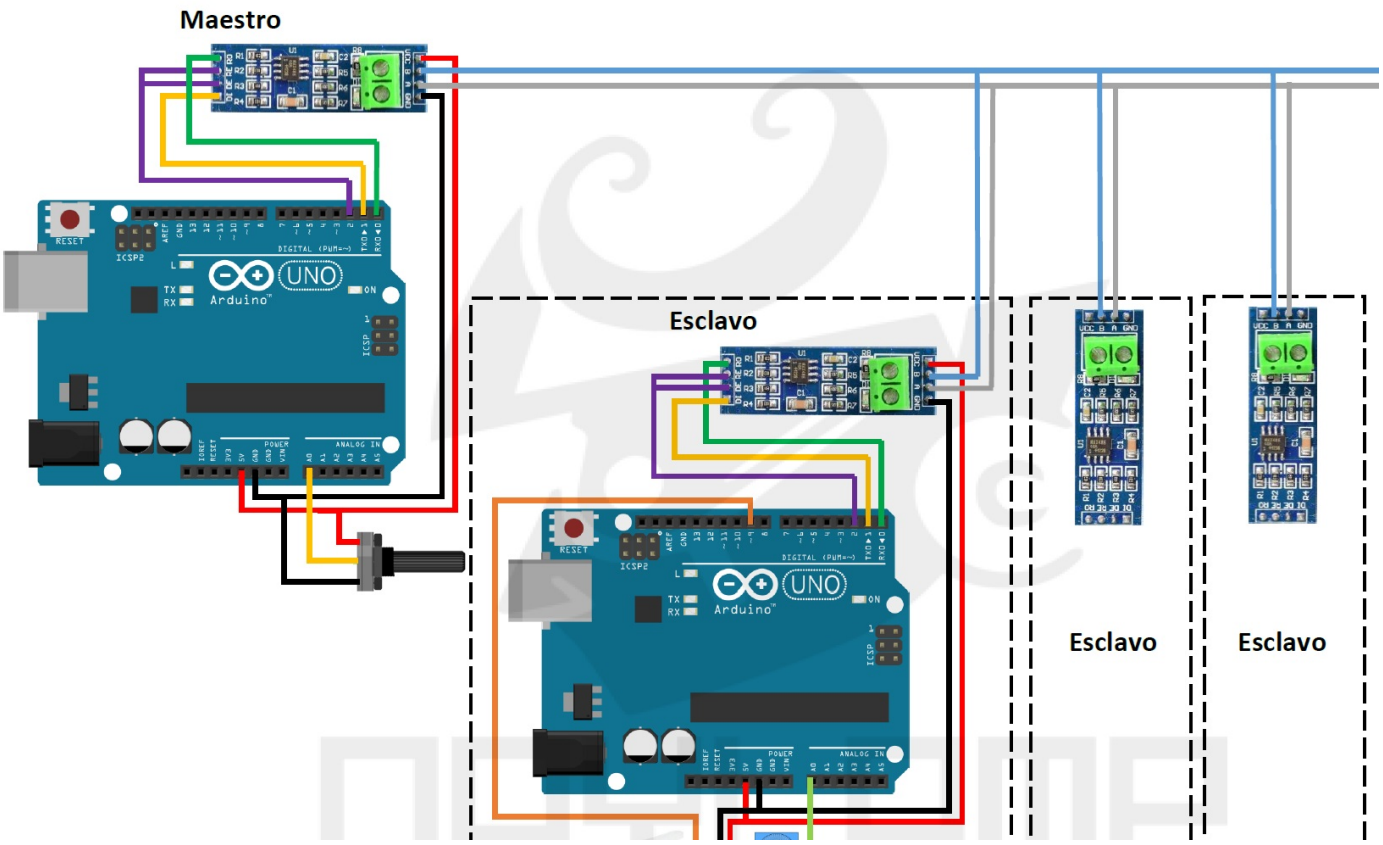
EL maestro siempre tiene activo la línea como transmisión pudiendo escribir en cualquier momento, mientras que el esclavo siempre está en modo recepción, escuchando los datos que le lleguen. Cuando el maestro necesite una respuesta por parte del esclavo, después de enviar la consulta, debe cambiar a modo de receptor para que el esclavo puede usar el canal para transmitir los datos, finalizado la respuesta, el maestro nuevamente debe tener libre el canal para transmitir.

Ej 4. Comunicación half dúplex entre varios arduinos.

Esta es la configuración más común que se usa, todos los Arduinos están conectados al mismo bus RS485, Un Arduino es maestro y todos los demás son Esclavos. Cada esclavo tiene una dirección el cual le identifica, el maestro para que pueda comunicarse con un esclavo usa esta dirección. El maestro envía la información por el bus y solo el esclavo con la dirección correcta es quien interpreta o procesa los datos.

A continuación mostramos el mismo ejemplo que se está trabajando anteriormente, pero enfocado une una conexión multipunto.

Las conexiones serían las siguientes





El ejemplo solo se muestra para un esclavo pero para los otros esclavos las conexiones son las mismas y en el código solo hay que tener en cuenta las direcciones de los esclavos.

La diferencia con los casos anteriores es en la trama para la comunicación, ahora es necesario enviar la dirección del esclavo, quedando la trama de la siguiente forma:

[Inicio de trama][Dirección][Función][Valor][Fin de trama]

La dirección es un número entero e indica con cual esclavo nos queremos comunicar. Ejemplos de esta trama serian: “I101S90F”,”I25S120F”,”I223LF”, etc.

El esclavo interpreta esta trama y si coincide su dirección entonces realiza la función correspondiente, si es una función que necesite responder, en la trama de respuesta también agrega su dirección, esto para que el maestro sepa que ha respondido el esclavo correspondiente. La trama de respuesta es de la forma:

[Inicio de trama][Dirección] [,] [Valor][Fin de trama]

En este caso entre la dirección y el valor enviamos una coma como separador puesto que ambos son valores numéricos, unos ejemplo de trama serian: “i101,865f”, “i26,64f”

Código del Maestro:

```
void setup()
{
  Serial.begin(9600);
  Serial.setTimeout(100); //establecemos un tiempo de espera de 100ms
  // inicializamos los pines
  pinMode(ledPin, OUTPUT);
  pinMode(EnTxPin, OUTPUT);
  digitalWrite(ledPin, LOW);
  digitalWrite(EnTxPin, HIGH); //RS485 como Transmisor
}

void loop()
{

  int lectura = analogRead(0);//leemos el valor del potenciómetro (de 0 a 1023)
  int angulo= map(lectura, 0, 1023, 0, 180);// escalamos la lectura a un valor de ángulo (entre 0 y
180)
  //---enviamos el ángulo para mover el servo------
  Serial.print("I"); //inicio de trama
  Serial.print("101");//dirección del esclavo
  Serial.print("S"); //función  S para indicarle que vamos a mover el servo
  Serial.print(angulo); //ángulo  o dato
  Serial.print("F"); //fin de trama
  //----------------------------
  delay(50);
  //---solicitamos una lectura del sensor----------
  Serial.print("I"); //inicio de trama
  Serial.print("101");//direccion del esclavo
  Serial.print("L"); //L para indicarle que vamos a Leer el sensor
  Serial.print("F"); //fin de trama
  Serial.flush();    //Esperamos hasta que se envíen los datos
  //----Leemos la respuesta del Esclavo-----
  digitalWrite(EnTxPin, LOW); //RS485 como receptor
  if(Serial.find("i")) //esperamos el inicio de trama
  {
      int esclavo=Serial.parseInt();  //recibimos la direccion del esclavo
      int dato=Serial.parseInt();  //recibimos el dato
      if(Serial.read()=='f'&&esclavo==101) //si fin de trama y direccion son los correctos
      {
         funcion(dato);   //realizamos la acción con el dato recibido
      }
  }
      digitalWrite(EnTxPin, HIGH); //RS485 como Transmisor
  //----------fin de la respuesta----------

}
void funcion(int dato)
{
```

```
  else
  digitalWrite(ledPin, LOW);
}
```

Código del Esclavo:

```
Servo myservo;  // creamos el objeto servo
const int EnTxPin =  2;  // HIGH:TX y LOW:RX
const int mydireccion =101; //Direccion del esclavo
void setup()
{
  Serial.begin(9600);
  Serial.setTimeout(100);  //establecemos un tiempo de espera de 100ms
  myservo.attach(9);  // asignamos el pin 9 para el servo.
  pinMode(EnTxPin, OUTPUT);
  digitalWrite(EnTxPin, LOW); //RS485 como receptor
}

void loop()
{
  if(Serial.available())
  {
    if(Serial.read()=='I') //Si recibimos el inicio de trama
    {
        int direccion=Serial.parseInt(); //recibimos la direccion
        if(direccion==mydireccion) //Si direccion es la nuestra
        {
            char funcion=Serial.read(); //leemos el carácter de función

            //---Si el carácter de función es una S entonces la trama es para mover el
motor-----------
            if(funcion=='S')
             {
                 int angulo=Serial.parseInt(); //recibimos el ángulo
                 if(Serial.read()=='F') //Si el fin de trama es el correcto
                 {
                   if(angulo<=180) //verificamos que sea un valor en el rango del servo
                    {
                      myservo.write(angulo); //movemos el servomotor al ángulo correspondiente.
                    }
                 }
             }
             //---Si el carácter de función  es L entonces el maestro está solicitando una lectura
del sensor---
             else if(funcion=='L')
             {
                if(Serial.read()=='F') //Si el fin de trama es el correcto
                 {
                   int lectura = analogRead(0); //realizamos  la lectura del sensor
                   digitalWrite(EnTxPin, HIGH); //rs485 como transmisor
                    Serial.print("i"); //inicio de trama
                   Serial.print(mydireccion); //direccion
                   Serial.print(",");
                   Serial.print(lectura); //valor del sensor
```

```
                digitalWrite(EnTxPin, LOW); //RS485 como receptor
              }
            }
          }
        }
      }
      delay(10);
}
```

El código para un segundo esclavo si se desea que tenga la misma función, en el código solo debe cambiar su dirección.

En un próximo tutorial implementaremos el protocolo Modbus en Arduino.

PRODUCTOS RELACIONADOS

En stock

Módulo MAX485 Conversor RS485 A Serial TTL

SKU: 000062

S/ 10,00

62 COMENTARIOS

Ed**** 01/01/2022 Responder

Buen día a todos, gracias por la información, pero tengo una duda en la comunicación simplex. Si necesito conectar dos o más controladores esclavos ¿debo conectar las líneas A y B del maestro en paralelo con los controladores esclavo?

Jo** ** ******do 15/11/2020 Responder

Buenas tardes y felicidades. He montado en mi segunda vivienda una instalación fotovoltaica y para controlar el vertido a red 0, necesito saber el consumo que se está produciendo en mi vivienda principal.
Como la instalación la tengo en una nave adyacente necesito enviarle la señal de consumo al inversor para que proceda a no inyectar nada de la producción fotovoltaica a la red.
He pensado a través de un emisor-receptor RS485 enviar los datos de consumo al inversor.
El problema está en que no tengo capacidad para programar los módulos LORAS correspondientes, de ahí, que solicite vuestra ayuda.
Lamento si no he logrado explicarme bien pero me brindo, a aclarar todo tipo de dudas que os puedan surgir. Saludo para todos.

Da*** 03/11/2020 Responder

Buenas noches, o primero felicitaros por la web. He estado mirando vuestro sketch. Mi pregunta es yo quiero hacer lo mismo pero con 4 potenciómetros y con 4 servos. Pero por mas que intento solo se mueve el primer servo. me podéis echar una mano?
Un saludo.

Al***** 23/07/2020 Responder

Hola,muy buen aporte como siempre, tienes algún ejemplo de esto pero usando alguna librería esto es muy útil si no es necesario comprobar los CRC , no ?

Fr******* 19/05/2020 Responder

Estoy intentando conectar un maestro con dos esclavos mediante hafl-duplex, cada uno de estos esclavos tienen sensores conectados. Ya hice todo el código para cada esclavo y el maestro, con todas las tramas. Ahora el problema es que solo me envía datos de un esclavo, claro, cada esclavo tiene direcciones diferentes. Alguien sabe porque o tiene la solución? Se lo agradeciera bastante :).
Saludos

de**** 14/12/2019 Responder

hola como hago el protocolo conitel con rs485

Gu******* ******* ******io 10/10/2019 Responder

Hola Naylamp, consulta: es posible reemplazar el medio fisico (par trenzado) por transmision wireless rf ? Gracias.

di********* 05/08/2019 Responder

Gracias por la informacion, ha sido muy util

Ju** 05/06/2019 Responder

Excelente! de gran ayuda

Na****** 09/05/2019 Responder

Muy buenos tutoriales... lástima no sacaron el de Modbus.

to********** 19/03/2019 Responder

Hola, respecto a la comunicació entre varios arduinos half dúplex, no veo como realizas la sincronización de los dispositivos , tanto maestro como esclavos. Para dos esclavos, seguramente funcionará de manera bastante fiable, pero al incrementar dicha cantidad , no habran continuos choques y perdidadas de información ?
Saludos

Mil gracias por tu aporte. Me podrías ayudar a hacer ésto mismo, pero con un arduino DUE como maestro?

Al** ****na 23/12/2018 Responder

Hola, primero que nada decir que el post es excelente, muy bien explicado y sencillo... solo me queda una duda, tanto el maestro como el esclavo deben tener su propia alimentación? o puede el maestro alimentar al esclavo directamente con la misma fuente, y que ese voltaje viaje a través de unos 20 o 40 metros de largo? gracias por su amable respuesta...saludos

Lu** ******** *****ez 18/03/2019 Responder

Claro que los puede alimentar así, pero lo más recomendable es alimentarlos independientemente

Jo** ***id 31/10/2018 Responder

Me ha encantado tu trabajo, pero tengo una duda si quiero realizar una conexion con 3 esclavos y que sea full duplex, me basaria en el esquema 2 o como se realizaria?

GE***** 18/09/2018 Responder

Genial. Claro. Conciso y con ejemplos.

Fr***** ****ll 16/08/2018 Responder

Buenas noches, tengo una situación algo extraña tengo conectados dos arduinos esclavos a un maestro y la información se envía correctamente, pero al conectar un tercer esclavo no funciona para nada, y los terminales (A y B) se hacen cortocircuito porque la resistencia disminuye mucho, qué puedo hacer para resolver esto?

Je*** ****ez 31/12/2017 Responder

Hola, muchas gracias por su gran trabajo, me ha servido de gran utilidad en mis proyecto. Ahora tengo un problemilla con un analizador de redes eléctricas de la serie CVM_C10. CIRCUTOR. Tiene un puerto de comunicación RS485 MODBUS RTU. y me gustaría leer la trama de datos con un Arduino y mostrarlo en un LCD. El problema es que no se como enviar la trama para preguntar y obtener los datos. gracias.

Hola, felicitaciones por su trabajo. esto es profesional
Intente implementar el uso del modulo pero no lo logre..por mas que busque información clara no la encontré
hasta dar con ustedes.
solo con la lectura de sus instrucciones , el concepto correcto del uso del modulo..
Gracias.

Ca**** 11/10/2017 Responder

Hola, tengo una pregunta: Pudiera utilizarse una lógica para usar el módulo tanto de tx como de rx manejando los pines DE y RE para cortocircuitarlos o alimentarlos en dependencia de si queremos usarlo de Tx o Rx

Da**** 03/09/2017 Responder

Hola, tengo una duda simple pero que no he podido resolver, este módulo me funcionaría con una tarjeta de desarrollo con microcontrolador Kinetis K64? lo pregunto porque es ARM y por ende trabaja a 3.3V y no a 5V como Arduino, me da miedo quemar mi placa.

Fr******* 03/08/2017 Responder

Hola, hay alguna manera de realizar una conexión full duplex entre varios arduinos?, muchas gracias

Ma****** 25/07/2017 Responder

Ya lo revisé era un error que tuve al modificar el código, muchas gracias!

Ma****** 25/07/2017 Responder

Hola, primero que nada muchas felicidades a Naylamp Mechatronics por sus tutoriales, realicé su ejemplo de Half Duplex entre 2 arduinos en comunicación bidireccional utilizando el ejemplo 4, la cuestión es que al monitorear el puerto serial del arduino maestro no se muestra la respuesta del arduino esclavo, esto es muy importante para mi pues estoy trabajando en un proyecto donde necesito esa comunicación bidireccional, por tu atenciòn muchas gracias.

Sw*** 15/12/2018 Responder

Hello Naylamp, your tutorial is very nice. I am using your code for Halfduplex between two Arduinos usng example 4,i am facing a problem that slave is not showing any response. Please help me actually I am working on a project which is on very high priority. In this project i need

please help me....

Jo*** 11/07/2017 Responder

Excelente aporte, podrian ayudarme mi cuestion es que la comunicacion funciona muy bien pero entre 2 arduinos, como hago para comunicar 5 esclavos con el master , el master va llamando uno por uno y esperando la repuesta de cada uno y ahi si llama al siguiente, pero no me funciona Agradezco de su ayuda, gracias...

Xa**** 30/01/2018 Responder

Hola, elimina la resistencia R7 (la de 120R) de los dispositivos que estén en medio de tu red. Los extremos SÍ DEBEN LLEVARLA.

Na***** 20/07/2017 Responder

Hola Jorge, tienes que identificar cuál de los esclavos está fallando o tal vez el maestro. Prueba uno por uno para tratar de identificar el error, también puedes conectar el RX de un conversor serial al RX o TX del Maestro, para monitorear las tramas de envío y respuesta.

Lu** 10/07/2017 Responder

Muy buen aporte estimado, voy a guiarme de tu ejemplo para hacer mis pruebas, solo como comentario, RS485 no es un protocolo, es un estandar de comunicación, protocolo es Modbus RTU, DeviceNet, etc...

Na***** 20/07/2017 Responder

Hola Luis, gracias por la observación, ya fue corregido.

Fr**** ******lo 28/06/2017 Responder

Hola amigos buen post, no quiero que me resuelvan las cosas pero estoy tratando de hacer una interfaz para manejar luces DMX a travez de wifi con protocolo arnet. He conseguido casi todo lo que no encuentro es un ejemplo de como conectar arduino uno y el modulo 485 y sus salidas a el canon.
Si tienes algun ejemplo les agradezco.

Go******** 22/03/2017 Responder

Gracias por las explicaciones.

quiero conectar varios arduinos con el metodo half duplex, mi pregunta es como podría acompañar este bus con otros 2 cables para alimentar todos los arduinos (alguno a 20 metros del principal / fuente de alimentación)

Gracias

Pu******* ****an 13/03/2017 Responder

Como puedo proteger la entrada de señal A-B para que no se me queme placa por descargas estaticas o electricas en la linea de conexion entre los modulos.
Sabes si hay que poner los arduinos o las plaquitas rs485 a tierra???

Xa**** 30/01/2018 Responder

Hola, podrías utilizar opto-acopladores, ya sean comunes o de alta velocidad. También podrían funcionar los transformadores de señal, y en este caso tendrías un aislamiento galvánico; en el primer caso el aislamiento es óptico.

Na***** 21/03/2017 Responder

Hola Pugliesso, puedes usar cable blindado o apantallado, es suficiente con conectar un extremo a tierra (case).

Gi*** 06/03/2017 Responder

Tema: protocolo Modbus en Arduino

hola muchas gracias por los aportes que brindas, tienes disponible ... con librerías ModBus??? un ejemplo que nos sirva...

Gracias de antemano.

Na***** 21/03/2017 Responder

Hola Ginno, aún no hemos experimentado con Modbus en Arduino, pero puedes usar la siguiente librería: https://github.com/4-20ma/ModbusMaster, cuya documentación está en: ModbusMaster

Hola que tal¡¡ Me interesa realizar algo parecido, pero mi intension, es comunicar un arduino con un plc delta, y desde el arduino activar bobinas, leer y modificar registros del PLC, no se si sea posible, Gracias.

ja** 06/10/2019 Responder

hola josue tambien quiero hacer lo mismo me gustaria contactarme con vos

La***** 20/12/2020 Responder

Hola!. También me interesa hacer lo mismo. ¿Pudieron llevarlo a cabo?.
Desde ya, muchas gracias

RA** ***GE 24/01/2017 Responder

excelente...¡¡gracias quisiera saber como puedo implementar el protocolo Modbus en Arduino.

Na***** 17/02/2017 Responder

Hola Raúl, el concepto es el mismo, puedes usar la siguiente librería:
https://github.com/4-20ma/ModbusMaster

Xa**** 30/01/2018 Responder

Hola. No se dice (ni se escribe) "librería"; el término correcto en español es "biblioteca".

An***** C. 15/01/2017 Responder

Saludos.
En primer lugar, enhorabuena y muchas gracias por este post y por los demás. Son claros, bien explicados y fáciles de seguir.
Estoy haciendo un proyecto en el que necesito activar unos reles desde un puesto central, distante unos 8 metros de estos. Comencé la construcción siguiendo tus explicaciones de la "Comunicación Half-Duplex entre Arduinos" y utilizando un Arduino UNO como Maestro y Arduino NANO en cada esclavo. Al conectar todo en el primero de ellos y probar todo fue perfecto. Sin embargo no he conseguido que funcione un segundo. He revisado y sustituido el cableado, he intercambiado las placas NANO, he intercambiado las placas RS485 y siempre he obtenido el mismo y frustrante resultado. El "puesto" que funciona SIEMPRE funciona (con cualquier placa) y el que no funciona NUNCA funciona. He instalado una pantalla que me confirma el comando enviado, incluyendo la dirección y, aunque se produce el pequeño "destello en ambas placas NANO de que algo ha llegado, el resultado siempre es el mismo. Hay algo adicional a comprobar o instalar? Gracias de antemano y

Ad** 22/03/2017 Responder

Que tal Antonio C. ya que toda la operación entre arduinos se realiza de forma correcta, me gustaría saber si tienes conectados los relevadores a una fuente externa y activados por transistor, ya que si los tienes directo a la arduino no tiene la corriente necesaria para activar la bobina.

Xa**** 30/01/2018 Responder

El tutorial no menciona algo SUMAMENTE IMPORTANTE: Los dispositivos extremos DEBEN incluir la resistencia de 120R (es decir, R7), mientras que los dispositivos de enmedio NO DEBEN incluirla. Además, me atrevería a recomendar QUITAR todas las resistencias de pull-up (R1..R4).

Esto es lo malo de Arduino, de los pseudo-expertos y de los aprendices que siempre serán aprendices: nadie se toma el tiempo suficiente para ENTENDER lo que se está haciendo. Desgraciadamente todo termina en un copy-and-paste, y a ver si con algo de suerte lo que sea que estemos haciendo funciona.

En este caso en particular, mi recomendación es leer sobre RS485 en libros y sitios en inglés, son los más completos y exactos.

BOLETIN
Regístrate en nuestro boletín para recibir las últimas ofertas y novedades.
Acepto los términos y condiciones y la política de privacidad
Tu correo electrónico

SOPORTE
Contactanos
Mapa del sitio
Envíos y Entregas
Métodos de Pago
Condiciones de uso

CATÁLOGO
Promociones especiales
Nuevos productos
¡Lo más vendido!

MI CUENTA
Mi cuenta
Mis ordenes
Mis direcciones

INFORMACIÓN DE CONTACTO
Urb. Ingeniería Mz-E Lt-31, Trujillo, Trujillo, La Libertad, Peru
contacto@naylampmechatronics.com
997646858
Horario de atención: Lun-Sab: 10am-7pm

Ad**** *******so 02/01/2017 Responder

Tengo el esquema armado y me anda perfecto. Entre placa y placa tienen un cableado de unos 50 a 70mts y funcionan bien. Estoy teniendo un problema los dias de tormenta siempre se me daña alguna plaquita RS485, es como si por el cable de datos A, B se metieras las descargas de la tormenta y le dañan algo a la placa (no la queman) pero es como si quedara transmitiendo datos erroneos. A Alguien le paso esto? como lo puedo solucionar?

Jo****** ****te 06/12/2016 Responder

Mejor y más claro de explicar imposible, un gran post. Pero tengo una cuestión, el módulo RS485 que ustedes proponen, trabaja con placas compatibles (no oficiales de Arduino). Gracias por el post!!!

Na***** 14/12/2016 Responder

Hola Joaquin, Si, le ejemplo trabaja con cualquier placa que soporte el IDE arduino como Arduino UNO, Nano, Mini, Micro, MEGA, ESP8266, NodeMCU etc.

©2021 Naylamp Mechatronics SAC

Columna