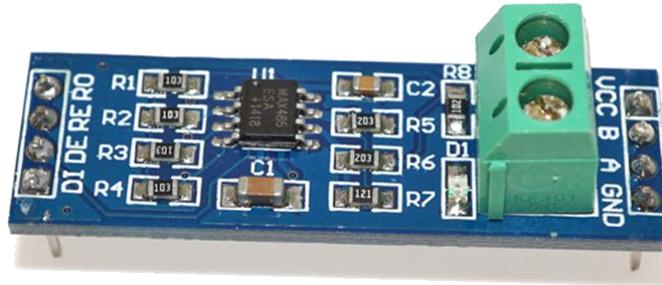


OKY3473: MODULO CONVERTIDOR TTL A RS485



Descripción

El sensor funciona a base del C.I MAX485, un transceptor de baja potencia y con una velocidad de respuesta limitada, es normalmente usado para comunicación RS-485. Puede alcanzar una velocidad máxima de transmisión de 2.5Mbps. Este módulo conecta a un microcontrolador o Arduino y permite convertir las señales seriales TTL a RS-485. RS485 se utiliza para comunicaciones serie a distancias más largas que el RS232 directo o TTL, y admite múltiples unidades en el mismo bus (Multi-Drop).

Una de las ventajas del RS485 es la larga distancia de transmisión. El alcance depende de la velocidad, siendo posible conseguir 35 Mbps en distancias inferiores a 10 metros, y hasta 100 Kbps en distancias hasta 1200 metros.

Aplicaciones

El RS485 se emplea frecuentemente en combinación con UARTs (Transmisor-Receptor Asíncrono Universal), para enviar la señales a largas distancias.

Algunos proyectos con RS485 incluyen automatización de plantas industriales. También se emplea en ámbitos de automoción e incluso en aviones para la conexión de dispositivos. Otros ejemplos de uso incluyen automatización de edificios, monitorización de sistemas fotovoltaicos, o control de grandes sistemas de iluminación o sonido como en conciertos de música.

Características

- Velocidad máxima de 10 Mbit/s (a 10 metros)
- Longitud máxima de alcance de 1200 metros (a 100 kbit/s)
- Bajo consumo de energía para la comunicación RS-485
- Tamaño: 44 mm x 14 mm.

Especificaciones

Voltaje de alimentación	5 V
Consumo de corriente máximo	500 uA (máx)
C.I de control	MAX485

PINOUT

Vcc	5VDC
A	Entrada no inversora del receptor y salida del controlador sin inversión
B	Entrada inversora del receptor y salida del controlador inverso
GND	Tierra
RO	Salida del receptor (al pin Rx del microcontrolador)
RE	Habilitación de salida del receptor (baja para habilitar)
DE	Activación de salida del controlador (alta para activar)
DI	Entrada del controlador (al pin Tx del microcontrolador)

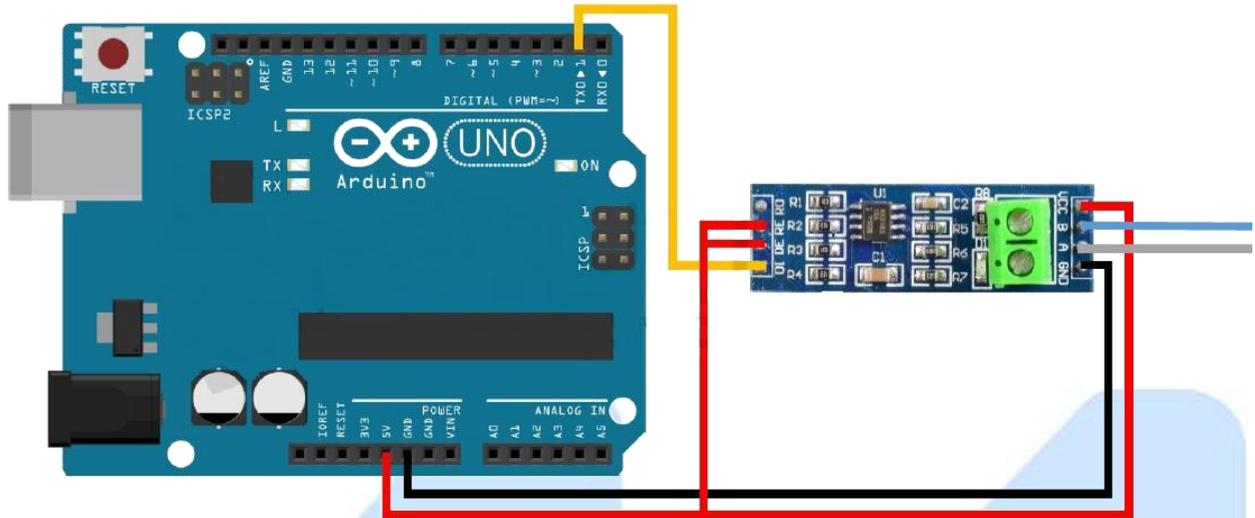
Diagramas de conexión y configuraciones básicas.

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. Los pines "A" y "B" se pueden conectar también desde la bornera.



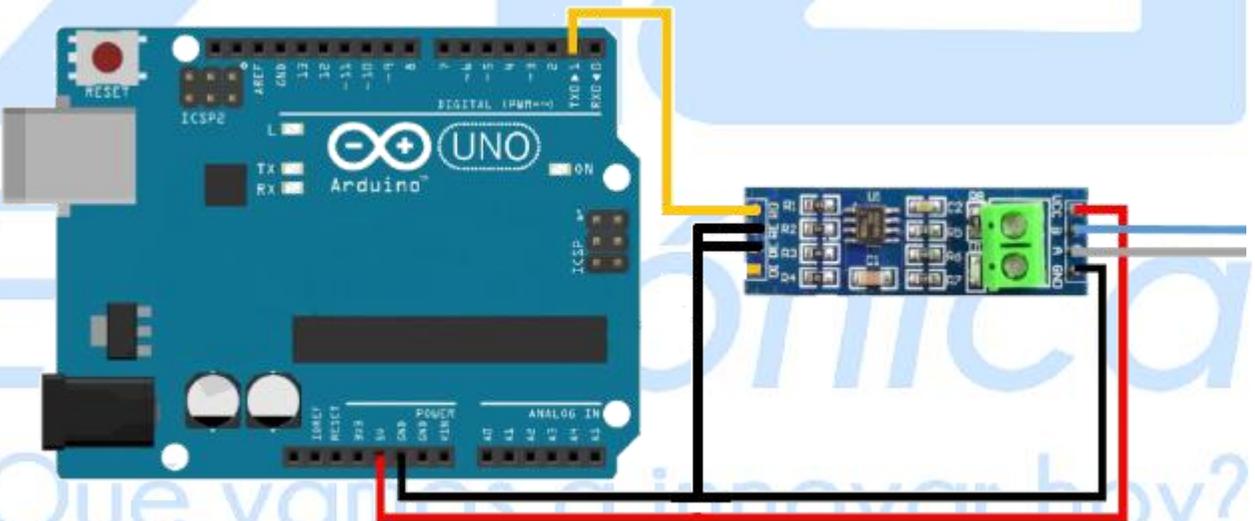
Usar el módulo como transmisor

En esta configuración el modulo solo trabaja como transmisor, para que el modulo sepa que las salida A y 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 como receptor

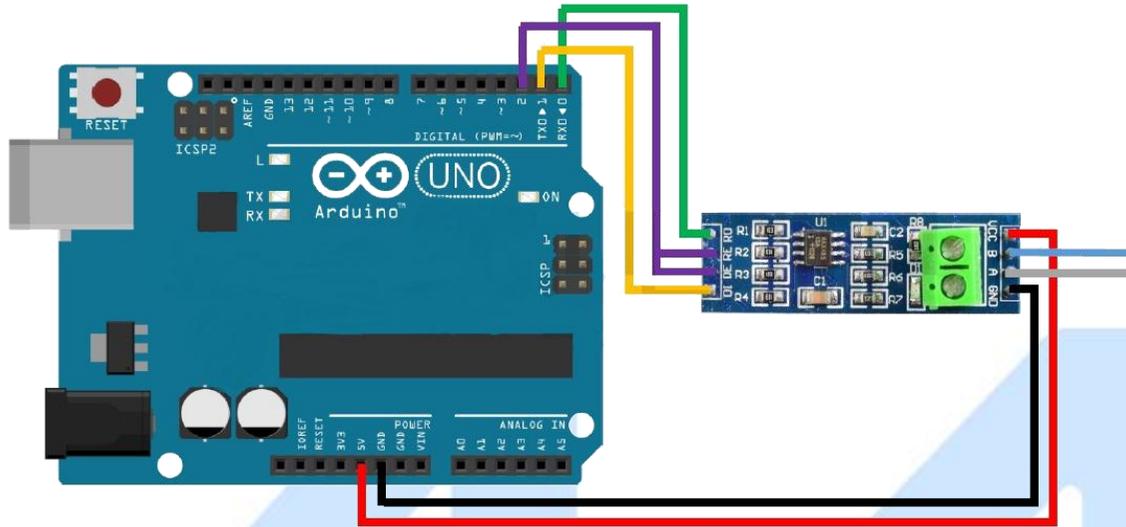
Al conectar los pines RE y DE el modulo se comporta como Receptor, y los datos recibidos por A B estarán presentes en el pin RO (Receiver Output), conectando el pin RO del módulo al RX de nuestro Arduino podemos leer los datos recibidos.



Usar el módulo como comunicación half dúplex

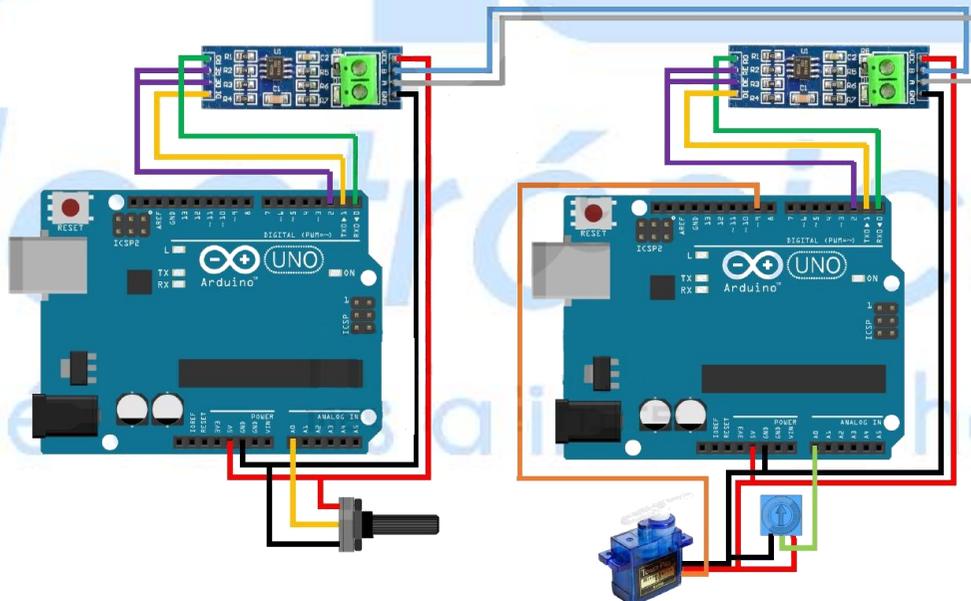
En una comunicación half dúplex utiliza un solo canal para comunicarse, es decir, cada Arduino puede actuar como emisor o receptor, pero no simultáneamente.

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.



Ejemplo

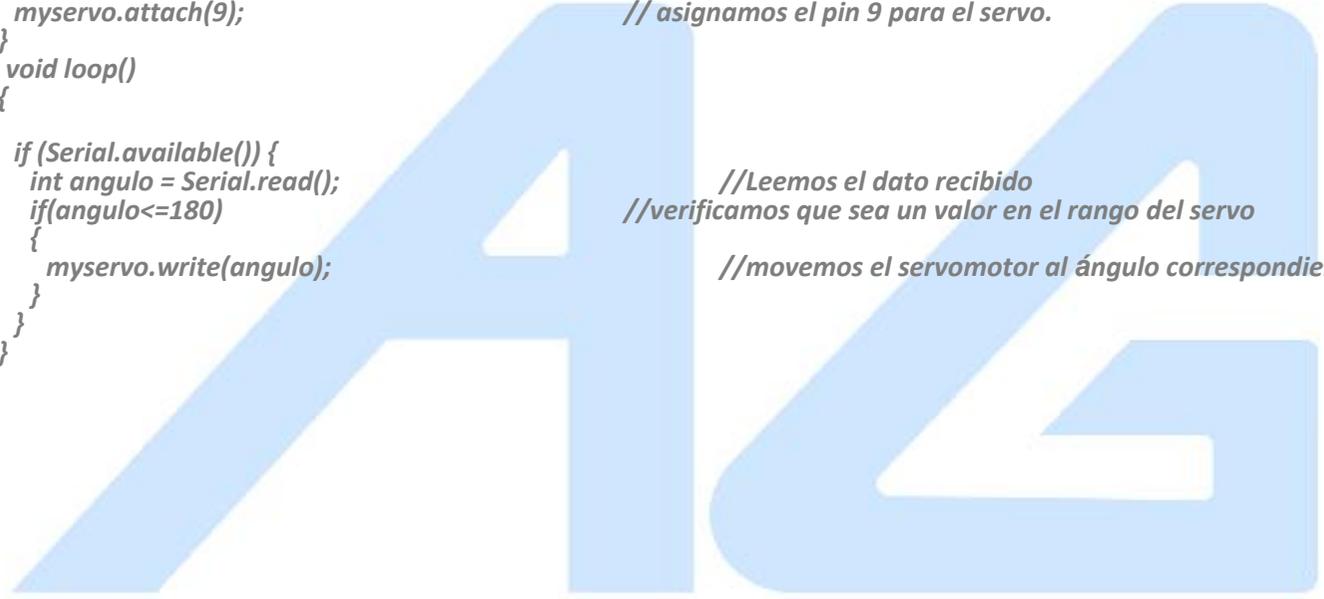
Este ejemplo muestra que a través de un potenciómetro conectado a un Arduino podemos mover un servomotor que estará conectado en otro Arduino, solo dos cables (salidas A y B del RS485) unirán a los dos Arduinos.



```
void setup()
{
  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);
}
```

Código del Arduino receptor

```
#include <Servo.h> // creamos el objeto servo
Servo myservo;
void setup()
{
  Serial.begin(9600);
  myservo.attach(9); // asignamos el pin 9 para el servo.
}
void loop()
{
  if (Serial.available()) { //Leemos el dato recibido
    int angulo = Serial.read(); //verificamos que sea un valor en el rango del servo
    if(angulo<=180) //movemos el servomotor al ángulo correspondiente.
    {
      myservo.write(angulo);
    }
  }
}
```



Electrónica
¿Qué vamos a innovar hoy?

	AG Electrónica S.A.P.I. de C.V. República del Salvador N° 20 Segundo Piso Teléfono: (01)55 5130 - 7210		
ACOTACIÓN: N/A	http://www.agelectronica.com	ESCALA: N/A	REALIZO: ILG REV: DGG
TOLERANCIA: N/A	MODULO CONVERTIDOR DE TTL A RS485		
TOLERANCIA: N/A	Fecha: 04/10/2019	No. Parte: OKY3473	