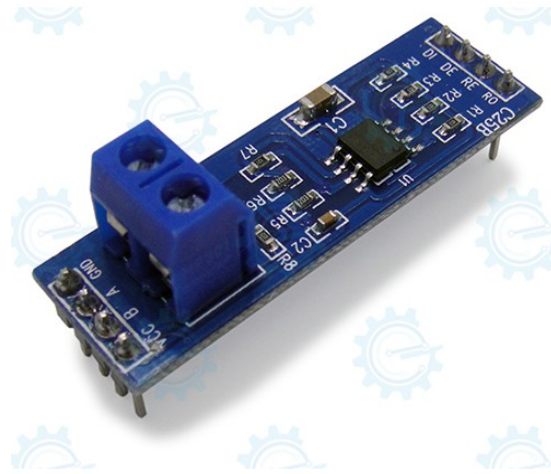


# MAX485 Module 5V logic TTL to RS-485 converter



This module interfaces an Arduino or similar microcomputer to RS-485. RS485 is used for Serial Communications over longer distances than direct RS232 or TTL, and supports multiple units on the same bus (Multi-Drop).

## General Specifications:

**Input Supply Voltage:** 5VDC

**Distance:** up to 1.2 km

**Type of cable:** straight/crossover

**Dimensions:** 44mm x 18mm

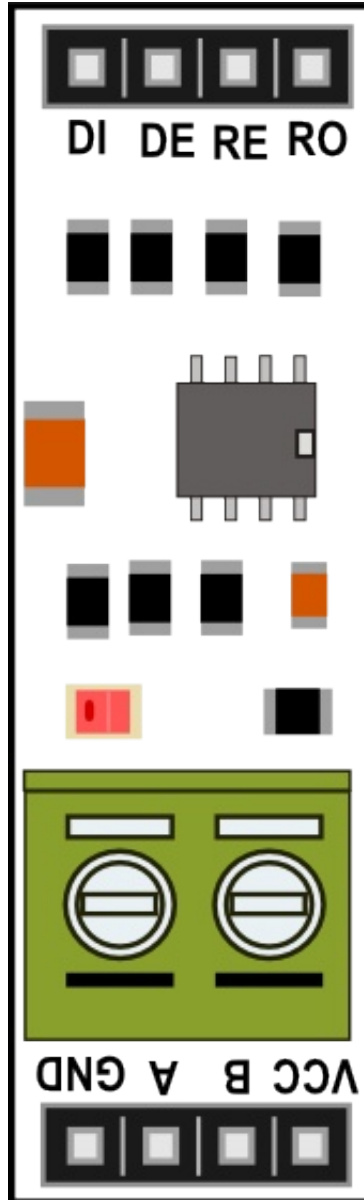


Figure 1. PCB Major Presentation

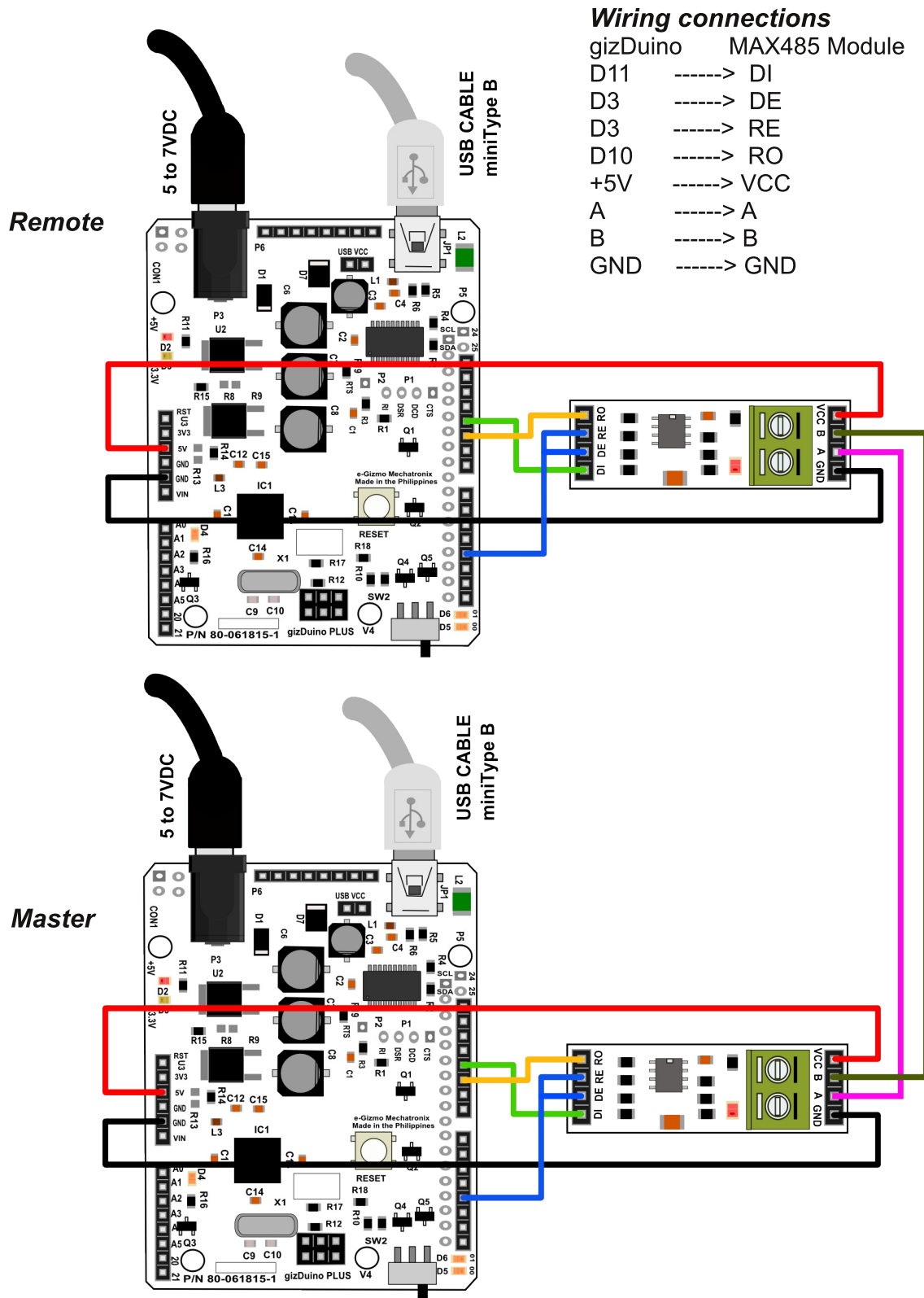


Figure 2. Sample connections

Upload this code to the gizDuino PLUS Microcontroller.  
then Open the Serial Monitor.

/\*

Master Sketch for MAX485 module 5v TTL to RS-485 converter

This sample code is for interface an Arduino used for  
Serial Communications over longer distances than direct RS232/TTL,  
Supports multiple units on the same bus.

Open Serial Monitor, type in top window.  
Should see same characters echoed back from remote Arduino.

Wiring Connections:

gizDuino board	RS-485 module
DI (Data In)	D11
RO (Receive Out)	D10
DE (Data Enable)	D3
and RE (receive enable)	
VCC	+5V
A	A (another RS-485)*
B	B *
GND	GND

- To other unit pins 11,10,GND (Cross Over)
- Pin 3 used for RS485 direction control

Reference:

<https://arduino-info.wikispaces.com/SoftwareSerialRS485Example>

Written by:

e-Gizmo Mechatronix Central

<http://www.e-gizmo.com>

March 23, 2017

\*/

```
#include <SoftwareSerial.h>

#define RXpin    10 //Serial Receive pin
#define TXpin    11 //Serial Transmit pin

#define TXcontrol 3 //RS485 Direction control
#define RS485Tx  HIGH
#define RS485Rx  LOW

#define LED      13
```

**Figure 3. Sample Code for Master**

```
SoftwareSerial RS485(RXpin, TXpin); // RX, TX

int byteReceived;
int byteSend;

void setup()
{
  Serial.begin(9600);
  Serial.println("USE SERIAL MONITOR, TYPE IN UPPER WINDOW, SEND!");
  pinMode(LED, OUTPUT);
  pinMode(TXcontrol, OUTPUT);
  digitalWrite(TXcontrol, RS485Rx);
  RS485.begin(4800); // set the data rate
}

void loop()
{
  digitalWrite(LED, HIGH);
  if (Serial.available())
  {
    byteReceived = Serial.read();

    digitalWrite(TXcontrol, RS485Tx); // Enable RS485 Transmit
    RS485.write(byteReceived);      // Send byte to Remote Arduino

    digitalWrite(LED, LOW); // Show activity
    delay(10);
    digitalWrite(TXcontrol, RS485Rx); // Disable RS485 Transmit
  }

  if (RS485.available()) //Look for data from other Arduino
  {
    digitalWrite(LED, HIGH); // Show activity
    byteReceived = RS485.read(); // Read received byte
    Serial.write(byteReceived); // Show on Serial Monitor
    delay(10);
    digitalWrite(LED, LOW); // Show activity
  }
}
```

```

/*
  Remote Sketch for MAX485 module 5v TTL to RS-485 converter

  This sample code is for interface an Arduino used for
  Serial Communications over longer distances than direct RS232/TTL,
  Supports multiple units on the same bus.

```

Remote received data and it loops back

Wiring Connections:

gizDuino board	RS-485 module
DI (Data In)	D11
RO (Receive Out)	D10
DE (Data Enable)	D3
and RE (receive enable)	
VCC	+5V
A	A (another RS-485)*
B	B *
GND	GND

- To other unit pins 11,10,GND (Cross Over)
- Pin 3 used for RS485 direction control
- Pin 13 LED blinks when data is received

Reference:

<https://arduino-info.wikispaces.com/SoftwareSerialRS485Example>

Written by:

e-Gizmo Mechatronix Central

<http://www.e-gizmo.com>

March 23, 2017

```
*/
```

```

#include <SoftwareSerial.h>

#define RXpin    10 //Serial Receive pin
#define TXpin    11 //Serial Transmit pin

#define TXcontrol 3 //RS485 Direction control
#define RS485Tx  HIGH
#define RS485Rx  LOW

#define LED      13

SoftwareSerial RS485(RXpin, TXpin); // RX, TX

int byteReceived;
int byteSend;

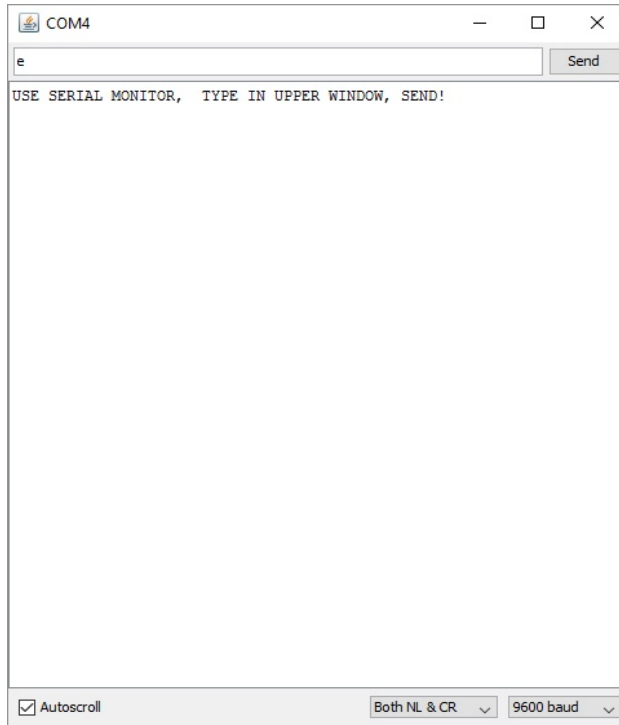
```

**Figure 4. Sample Code for Remote**

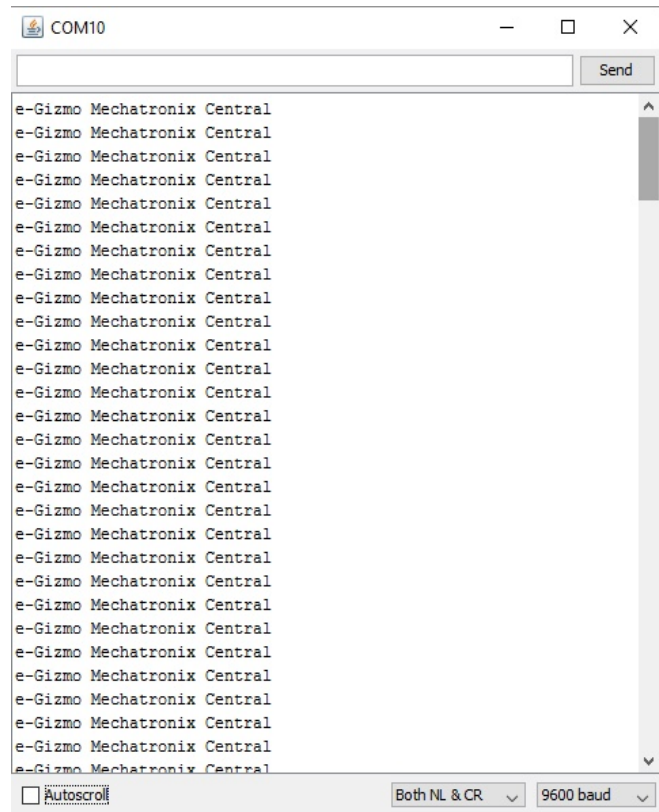
```
void setup()
{
  Serial.begin(9600);
  pinMode(LED, OUTPUT);
  pinMode(TXcontrol, OUTPUT);
  digitalWrite(TXcontrol, RS485Rx);
  RS485.begin(4800); // set the data rate
}

void loop()
{
  if (RS485.available())
  {
    byteSend = RS485.read();
    Serial.println("e-Gizmo Mechatronix Central");
    digitalWrite(LED, HIGH);
    delay(10);
    digitalWrite(LED, LOW);

    digitalWrite(TXcontrol, RS485Tx);
    RS485.write(byteSend);
    delay(10);
    digitalWrite(TXcontrol, RS485Rx);
  }
}
```



**Figure 5. Serial Monitor from master**



**Figure 6. Serial Monitor from the remote**