



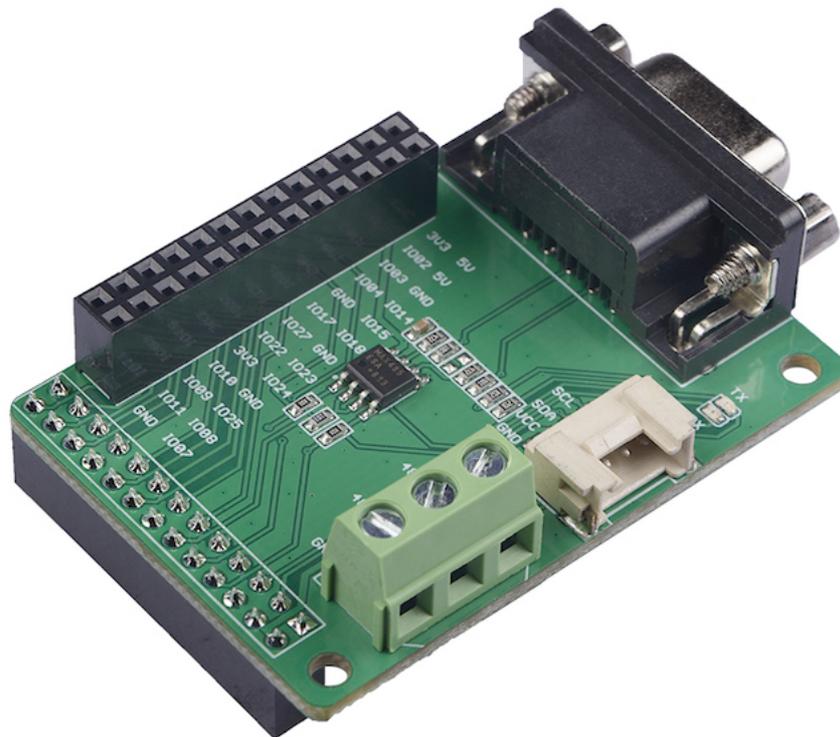
# RS-485 Shield for Raspberry Pi

Release date : 01/09/2019

Version : 1.0

Wiki: [http://wiki.seeedstudio.com/RS-485\\_Shield\\_for\\_Raspberry\\_Pi/](http://wiki.seeedstudio.com/RS-485_Shield_for_Raspberry_Pi/)

Bazaar: <https://www.seeedstudio.com/RS-485-Shield-for-Raspberry-Pi.html>



RS-485 is a cost-effective solution in serial communication networks. It can be used up to data rates at 10 Mbit/s or distances up to 1200m at lower speeds. This RS-485 Shield is a standard add-on board for Raspberry Pi. It is integrated with simple screw terminal as well as DB9 interface.

[Get One Now !\[\]\(0f848bbd71cef6b345273b16f905912a\_img.jpg\)](#)

## Feature

- One driver and one receiver per part
- EMI noise minimization
- Transmission rate up to 2.5Mbps
- No driver slew rate limitation
- Short-circuit current limited
- Fail-Safe Applications

## Specification

<b>Item</b>	<b>Value</b>
Operating Supply Voltage	3.3V
Interface	RS-485 DB9 Interface X1 RS-485 Screw Interface X1 2×13 Female Header To Raspberry X1 2×13 Expansion Female Header X1 Grove I2C Interface X1
Data Rate	2.5Mbps
Number of Receivers on	32
Storage Temperature Range	-65~160°C
Channel Number	8
Resolution	12-Bit
Power Consumption	Different power consumption depending on the transmission rate
Weight	42g
Length	62.0mm
Width	39.2mm
Height	21.8mm

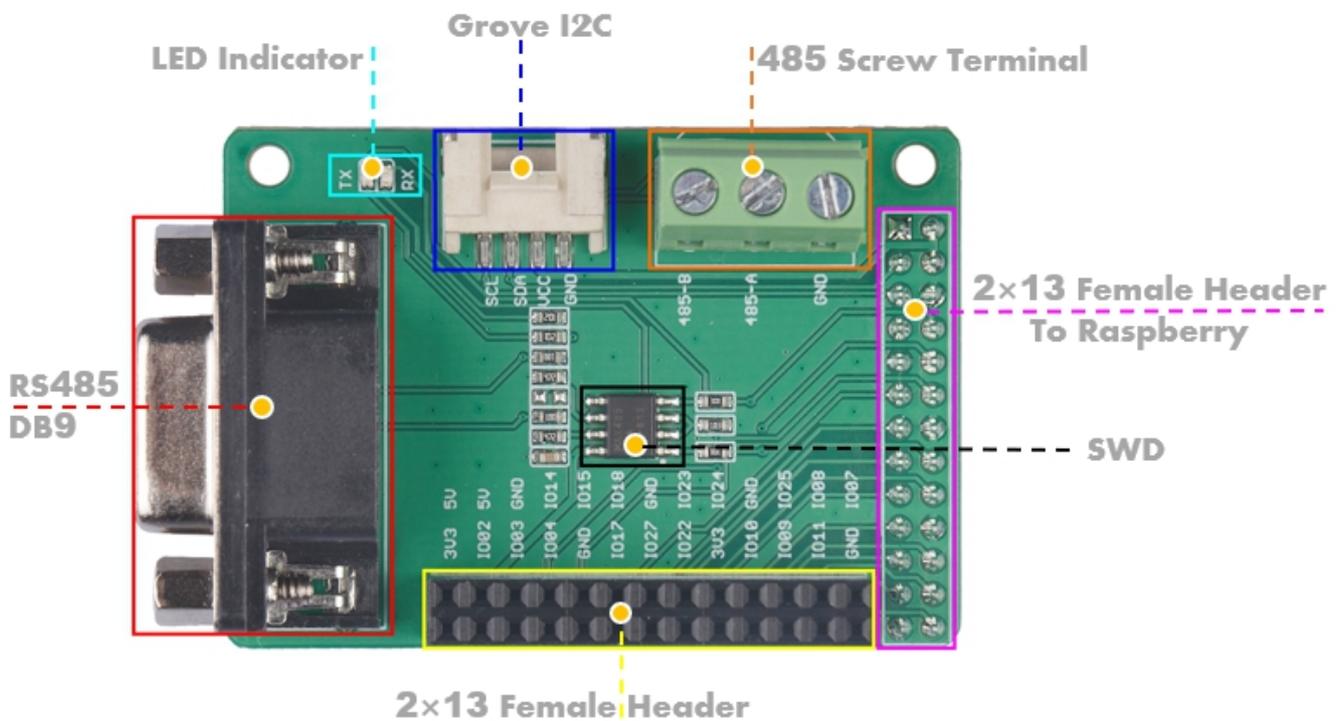
## Typical Applications

- Low-Power RS-485 Transceivers
- Level Translators
- Transceivers for EMI-Sensitive Applications Industrial-Control Local Area Networks
- Half-duplex applications

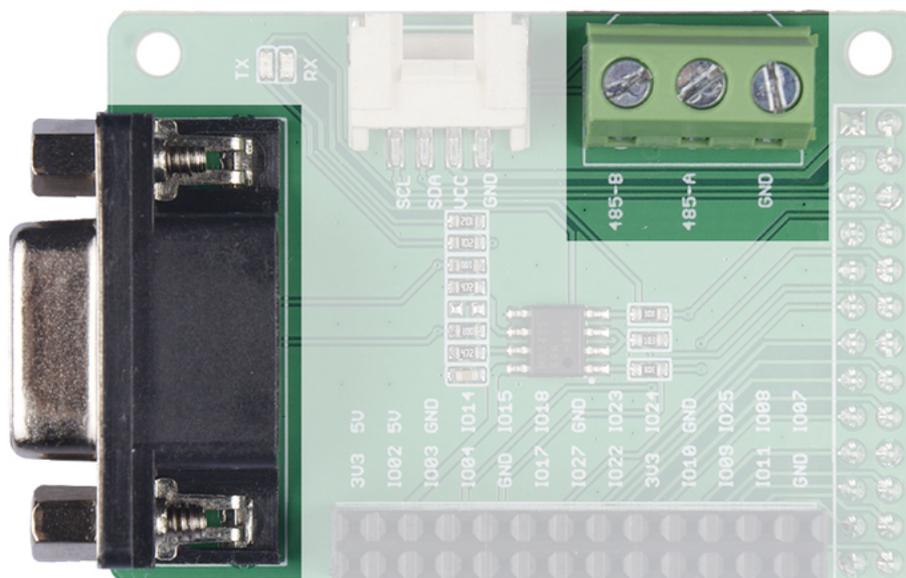
## Hardware Overview

### Pin Out

### Overview



## RS-485 DB9 Interface & RS-485 Screw Interface

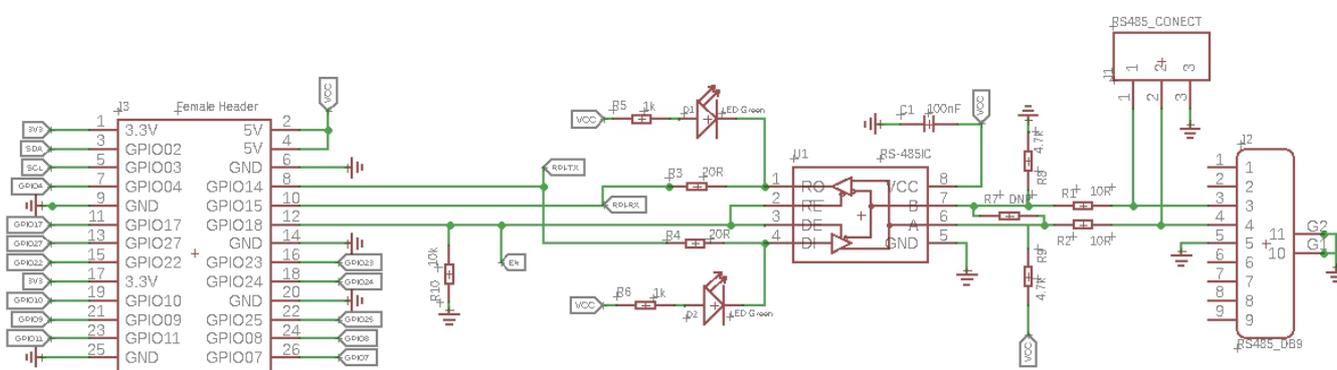


The 485 interface uses differential signal transmission. Please make sure the port A connect to the port A of the 485 device, and port B connect to the port B of the 485 device.

485-A: The end A of RS485 data transmit wire, connect to the pin A of the MAX485 chip.

485-B: The end B of RS485 data transmit wire, connect to the pin B of the MAX485 chip.

GND: Connect to the Raspberry Pi GND.



[You can click the picture to view the original file](#)

As you can see, GPIO14 and GPIO15 are applied for data transmission, and we use GPIO18 as the enable signal.

For the definition of logic signals, please refer to the table below.

**Table 1. Transmitting**

INPUTS			OUTPUTS	
$\overline{RE}$	DE	DI	Z	Y
X	1	1	0	1
X	1	0	1	0
0	0	X	High-Z	High-Z
1	0	X	High-Z*	High-Z*

*X = Don't care*

*High-Z = High impedance*

*\*Shutdown mode for MAX481/MAX483/MAX487*

**Table 2. Receiving**

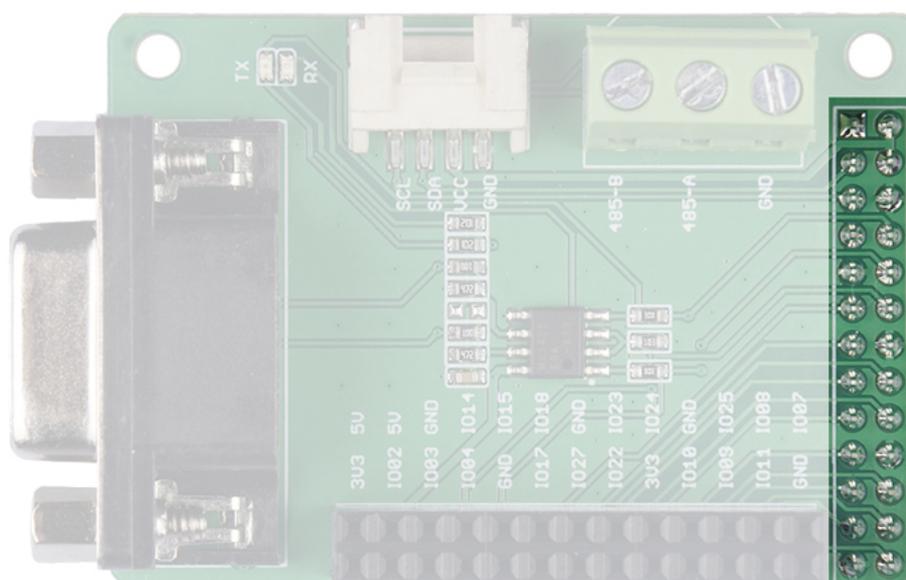
INPUTS			OUTPUT
$\overline{RE}$	DE	A-B	RO
0	0	$\geq +0.2V$	1
0	0	$\leq -0.2V$	0
0	0	Inputs open	1
1	0	X	High-Z*

*X = Don't care*

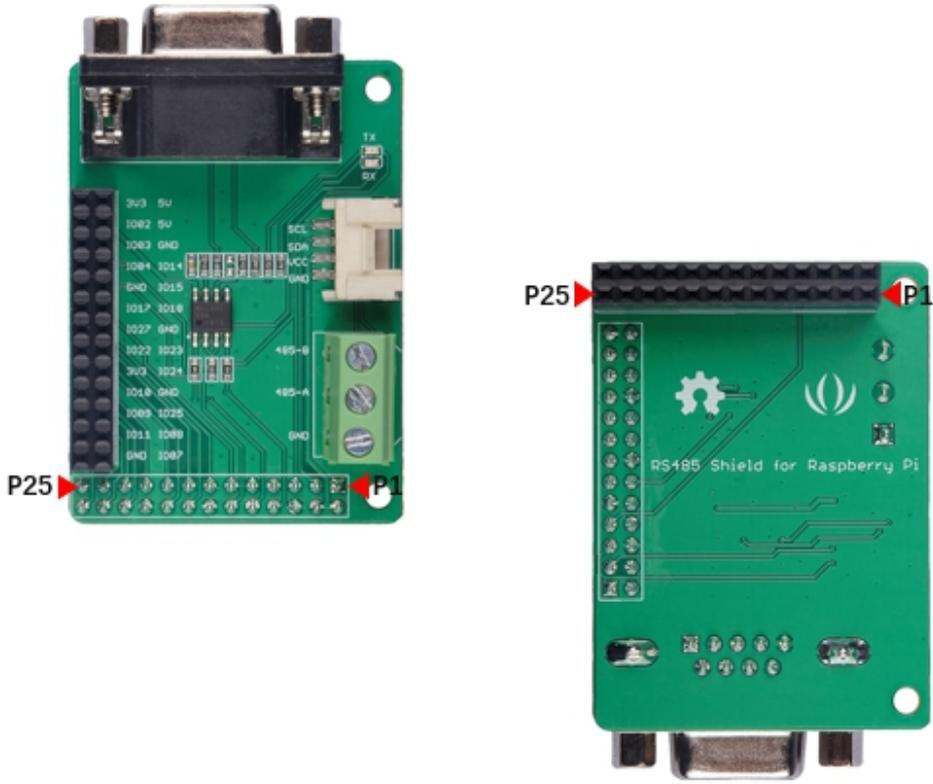
*High-Z = High impedance*

*\*Shutdown mode for MAX481/MAX483/MAX487*

## Female Headers to Raspberry

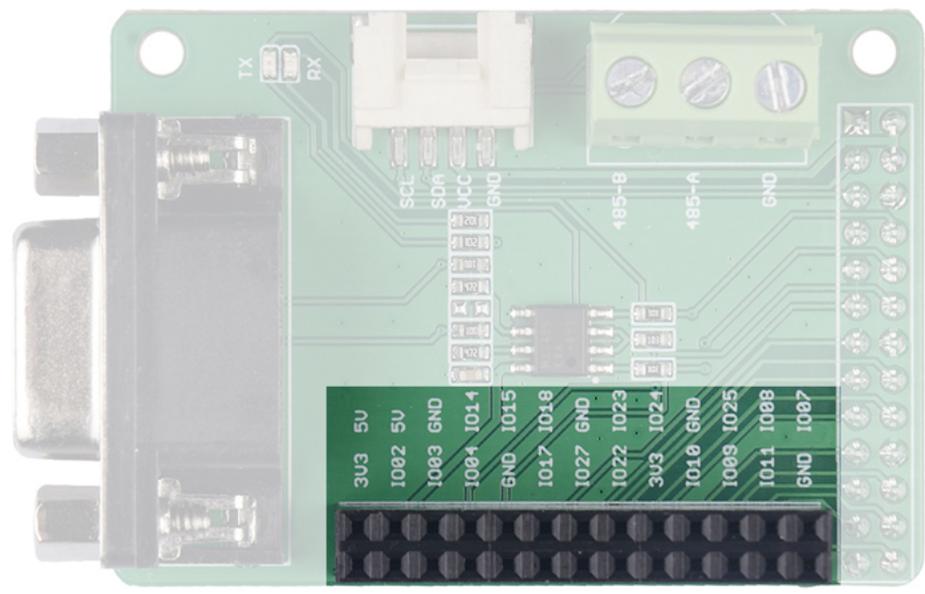


We use a 2X13 Female Header to plug this module into the Raspberry Pi, please make sure the pins are aligned.



---

### Extension Female Headers

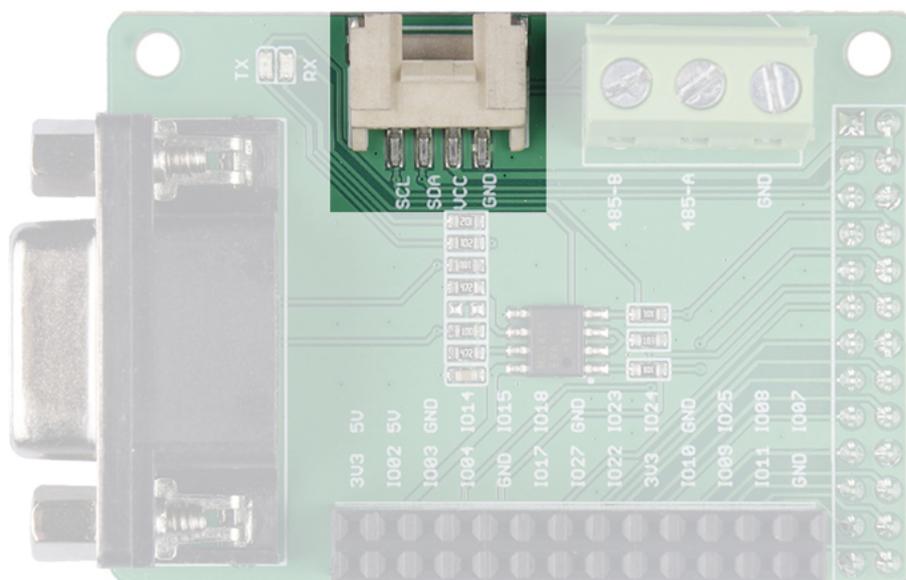


This RS-485 Shield occupies 26 Raspberry Pi pins, and only 5 GPIO pins are actually used. We take these 26 pins out in case you need those pins for other purposes.

## GPIO Occupied List

GPIO Number	Function
GPIO02	SDA for Grove I2C Port
GPIO03	SCL for Grove I2C Port
GPIO14	Connect to the pin <b>DI</b> of Max485 chip, for data transmission.
GPIO15	Connect to the pin <b>RO</b> of Max485 chip, for data transmission.
GPIO18	Connect to the pin <b>RE &amp; DE</b> of Max485 chip, work as the enable signal.

## Grove I2C Port



We have reserved the I2C interface so that you can easily use it with I2C devices. It should be noted that the VCC of this port is 5V, you need to confirm that whether the module is compatible with 5V voltage.

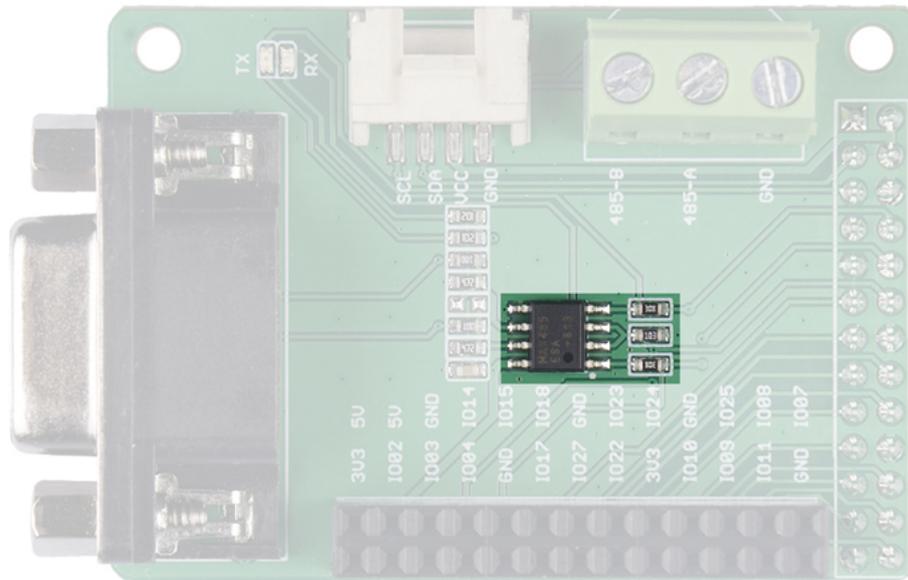
SCL: I2C serial clock, connect to the GPIO03 of Raspberry Pi.

SDA: I2C serial data, connect to the GPIO02 of Raspberry Pi.

VCC: Connect to the Raspberry Pi 5V pin.

GND: Connect to the Raspberry Pi GND pin.

## Max485 Chip



We use the MAX485ESA IC for this shield, for more detail about this IC, please check the [MAX485 Datasheet](#)

## Platforms Supported

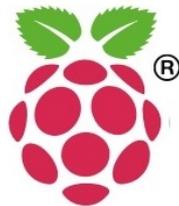
**Arduino**

**Raspberry Pi**

**BeagleBone**

**Wio**

**LinkIt ONE**



## Getting Started

### Hardware

#### Materials required

##### Raspberry pi



##### RS-485 Shield for Raspberry Pi



---

[Get ONE Now](#)

[Get ONE Now](#)

- **Step 1.** Plug the RS-485 Shield for Raspberry Pi into the Raspberry Pi.
- **Step 2.** Connect the **485-A** to 485 wire A, connect the **485-B** to 485 wire B.

!!!Note If the wire is reversed, communication will not be possible.

- **Step 3.** Power the Raspberry Pi with micro-usb cable.

### Software

#### Communication test code

You can create a new python file and copy the following code into the new file, or you can download the source file in the resource download area. Then run it in your terminal.

## Sending Code.

```
#!/usr/bin/env python

import time
import serial
import os

send_str = "*****rs485888888--\r\n"

os.system("echo 18 > /sys/class/gpio/export")
os.system("echo out > /sys/class/gpio/gpio18/direction")
ser = serial.Serial(port='/dev/ttyAMA0',baudrate
=1000000,bytesize=8,stopbits=1,timeout=1)

last_time = time.time()

now_time = time.time()
os.system("echo 1 > /sys/class/gpio/gpio18/value")
time.sleep(0.01)
n = 800
while n>0:
    ser.write(send_str)
    n=n-1
    # time.sleep(0.001)
os.system("echo 0 > /sys/class/gpio/gpio18/value")
```

## Receive Code

```
#!/usr/bin/env python

import time
import serial
import os

send_str = "*****abcdefghijklmnopqrstuvwxyz&"

os.system("echo 18 > /sys/class/gpio/export")
os.system("echo out > /sys/class/gpio/gpio18/direction")

ser = serial.Serial(port='/dev/ttyAMA0',baudrate
=115200,bytesize=8,stopbits=1,timeout=1)

last_time = time.time()
while 1:
    now_time = time.time()
    if((now_time-last_time)>=1):
        last_time = now_time
#        print "172 sending"
        os.system("echo 1 > /sys/class/gpio/gpio18/value")
        time.sleep(0.01)
        ser.write(send_str)
        time.sleep(0.01)
        os.system("echo 0 > /sys/class/gpio/gpio18/value")
os.system("echo 0 > /sys/class/gpio/gpio18/value")
time.sleep(0.01)
count = ser.inWaiting()
if(count != 0):
    x=ser.readline()
    if "*****" in x:
#        print "str length is: " + str(count)
        print x
```

You need two shield and two raspberry to test the code above, or you can use the serial tool in the PC to communicate with your raspberry pi.

## Resources

- **[Zip]** [RS-485 Shield for Raspberry Pi Eagle Files](#)
- **[Zip]** [Python Test Code](#)
- **[PDF]** [MAX485 Datasheet](#)
- **[PDF]** [PDF Format Wiki](#)

## Tech Support

Please do not hesitate to submit the issue into our [forum](#) or drop mail to [techsupport@seeed.cc](mailto:techsupport@seeed.cc).