QUICK REFERENCE · RS485

RS485 Board, MAX485

UART to RS485 for 5 V microcontrollers. Its twin, the MAX3485 board, is for 3.3 V ones.

5 V board: 5 V microcontrollers only

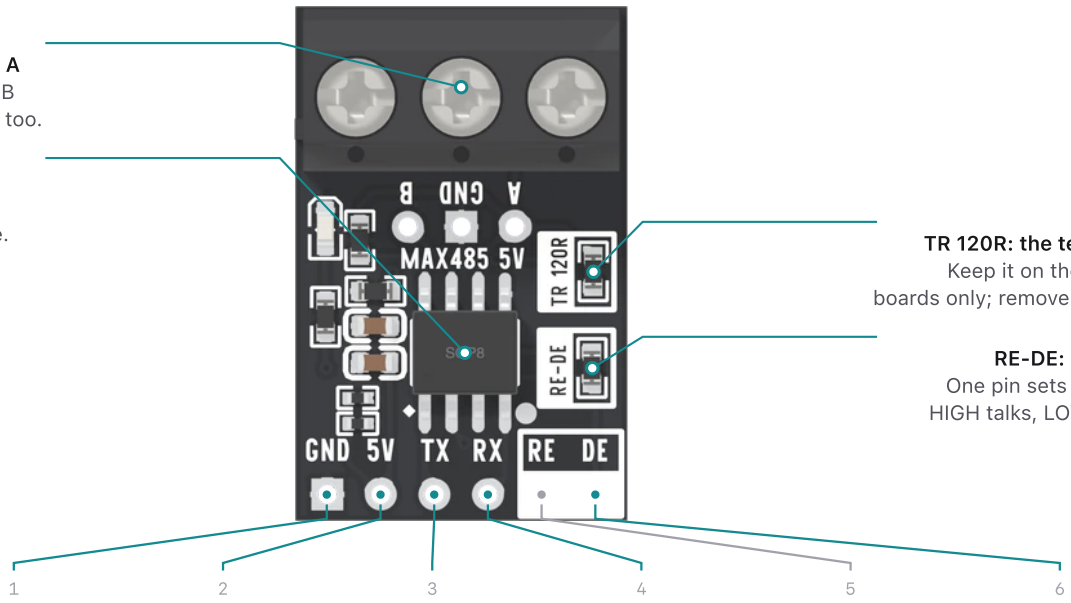
Its TX swings to 5 V and is pulled up to 5 V on the board, which a 3.3 V pin is not built to take. **On an ESP32, a Pico or any 3.3 V microcontroller, use the MAX3485 board instead.**

Pins, and which to wire

Drawn from above, terminal at the top. The six holes along the bottom go to your microcontroller; B, GND and A go to the bus, on the terminal or the three holes under it. From underneath the order reads mirror-image.

Screw terminal: B, GND, A
To the next board: A and B on one twisted pair, GND too.

MAX485, runs at 5 V
Half duplex: one board on the pair talks at a time.



TR 120R: the terminator
Keep it on the two end boards only; remove between.

RE-DE: a 0 Ω link
One pin sets direction: HIGH talks, LOW listens.

Front	**GND**	**5V**	**TX**	**RX**	**RE**	**DE**
Back	GND	5V	TX	RX	RE	DE
Your board	GND	5V	RX	TX	optional	DIR

TX goes to your RX The board names its pins from its own side. Its TX is the chip's receiver output, what it heard on the bus, so it goes to your RX; its RX takes your TX. TX to TX moves nothing in either direction, with no error anywhere.

The 120 Ω, on how many boards

The two end boards

60 Ω across A and B

25 mA to make 1.5 V: inside the 54 Ω the chips are specified into.

3 fitted

40 Ω across A and B

38 mA to make 1.5 V: past the 54 Ω the chips are specified into.

All twelve fitted

10 Ω across A and B

150 mA to make 1.5 V: past the 54 Ω the chips are specified into.

Every board ships with its 120 Ω fitted. Keep it on the two boards at the ends of the cable and take it off every board between: both chips are specified into 54 Ω, and three or more fitted is past it.

Arithmetic, not measurements: n resistors of 120 Ω in parallel, the current a driver needs for the 1.5 V minimum in both datasheets, and the idle difference the 20 kΩ bias pair leaves across the terminators at 5 V, with the receivers' own inputs left out. A receiver is sure of a 1 only above 200 mV, so a terminated bus with nobody talking is undecided: frame your messages and ignore stray bytes.

Two Unos, ping and pong

Arduino IDE · Board: Arduino Uno · SoftwareSerial, built in

```
// Same sketch on both Unos: TALKER true on one, false on the other.
#include <SoftwareSerial.h>
const bool TALKER = true;
const int DIR = 2;
SoftwareSerial bus(10, 11);  // RX, TX
String line;
int count = 0;
unsigned long last = 0;

void send(const String &msg) {
  digitalWrite(DIR, HIGH);      // take the bus
  delayMicroseconds(200);
  bus.println(msg);
  bus.flush();
  digitalWrite(DIR, LOW);       // let go
}

void setup() {
  pinMode(DIR, OUTPUT);
  digitalWrite(DIR, LOW);       // listen first
  Serial.begin(9600);
  bus.begin(9600);
}

void loop() {
  if (TALKER && millis() - last >= 1000) {
    last = millis();
    send("ping " + String(++count));
  }
  while (bus.available()) {
    char c = bus.read();
    if (c != '\n') { if (line.length() < 30) line += c; continue; }
    line.trim();
    Serial.println(line);
    if (!TALKER && line.startsWith("ping "))
      send("pong " + line.substring(5));
    line = "";
  }
}
```

```
pong 1
pong 2
```

The talker's Serial Monitor at 9600; the listener's prints ping 1, ping 2. Every pong crossed the pair twice.

Wiring

	FRONT	EACH UNO
1	**GND**	GND
2	**5V**	5V
3	**TX**	10
4	**RX**	11
5	**RE**	—
6	**DE**	2

Each Uno to its own board, then the two terminals A to A, B to B, GND to GND. Keep both 120 Ω fitted: two boards are the two ends. RE is the same net as DE, so it needs no wire. The Uno's pins 0 and 1 are its USB link, so the bus uses SoftwareSerial.

RE and DE, one pin

A 0 Ω link in the box printed RE-DE joins RE to DE, so one GPIO sets the direction: HIGH talks, LOW listens, never both. Nothing on the board pulls it either way, so set it LOW first thing in setup(), and drop it only after flush() returns, or the last byte is cut off.

When it does not work

Nothing arrives

TX to TX is the usual cause: the board's TX goes to your RX. Then check the talker raises its direction pin to send.

I read back my own messages

The receiver is on while you send: the RE-DE link was removed and RE tied LOW. Refit the link, or give RE the DE pin.

Stray bytes when nobody talks

Normal on a terminated bus. Send whole lines and ignore anything that is not one.

Every character is wrong

A and B are crossed between the terminals, which turns every bit over. Swap them at one end, and check both ends use one baud rate.

Fine on the desk, not on the cable

Run GND with A and B, twist A and B together, and keep the 120 Ω on the two end boards only.

It broke when I added boards

Each board brought its own 120 Ω. Measure A to B with the power off: about 60 Ω is right, under 50 means too many.

Specifications

Chip	MAX485 (UMW MAX485ESA)	**Supply and logic**	4.75 to 5.25 V
Fastest	2.5 Mbit/s	**Termination**	120 Ω across A and B, on every board
Bias	20 kΩ A to VCC, B to GND	**Direction**	RE and DE joined by a 0 Ω link
Receivers	Up to 32 on one bus, one unit load each	**Board**	15.76 × 24.42 mm, headers loose