

Roteiro de Teste de Funcionamento do Módulo conversor TTL para RS485 Arduino

O vídeo com o demonstrativo do Teste de Funcionamento do Módulo conversor TTL para RS485 Arduino pode ser acessado [clikando aqui!](#)



O Módulo conversor TTL para RS485 é um módulo utilizado para conversão de dados TTL para o protocolo RS485, ou RS485 para TTL. No contexto da utilização do módulo junto a embarcados como Arduino, o módulo se mostra bastante eficiente, permitindo a conexão de vários periféricos conectados ao sistema, podendo atuar com vários escravos em uma conexão, juntamente a uma alta velocidade de transmissão de dados. Além de atuar com grande performance em ambientes ruidosos e de longas distâncias.

O Módulo conversor TTL para RS485 Arduino é amplamente utilizado em aplicações em que são necessárias a transmissão de comunicação de um Arduino em linguagem TTL para um protocolo RS485, ou vice e versa, podendo atuar em aplicações de controle industrial e sistemas de automação.

Principais elementos que compõe o Módulo conversor TTL para RS485 Arduino



Pino VCC – +5V Alimentação Positiva

Pino B – Inversor da Entrada do Receptor e a Saída do Driver

Pino A – Entrada do Receptor Não Inversor e Saída do Driver

GND – Terra

Borne Comum a A e B – Pontos A e B comum aos pinos A e B

LED Indicador – Led Indicador de funcionamento

DI – Entrada do Driver

DE – Ativação da Saída do Driver

RE – Habilitação de Saída do Receptor

RO – Saída do Receptor

Chip MAX485 – Chip conversor TTL para RS485, composto

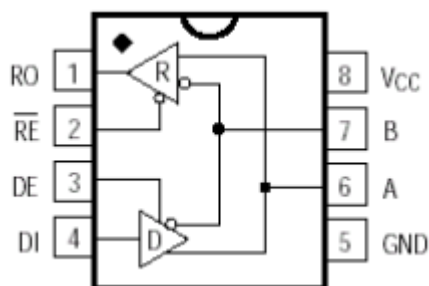
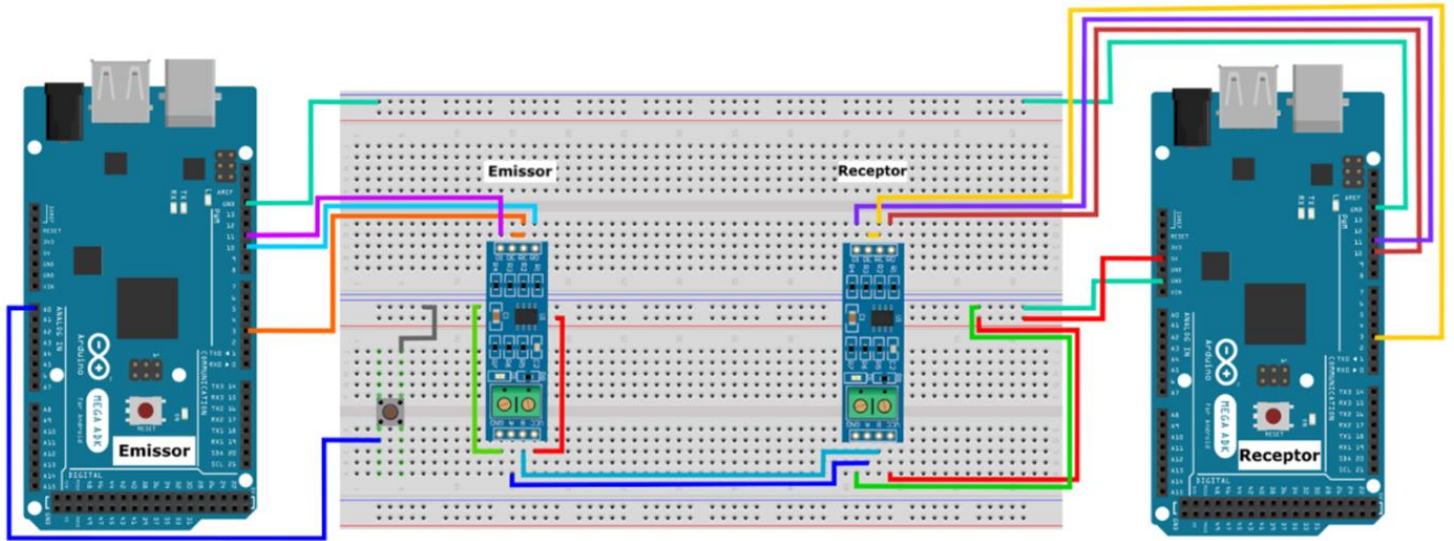


Diagrama eletrônico dos Módulos conversores TTL para RS485, conectados como emissor e receptor respectivamente



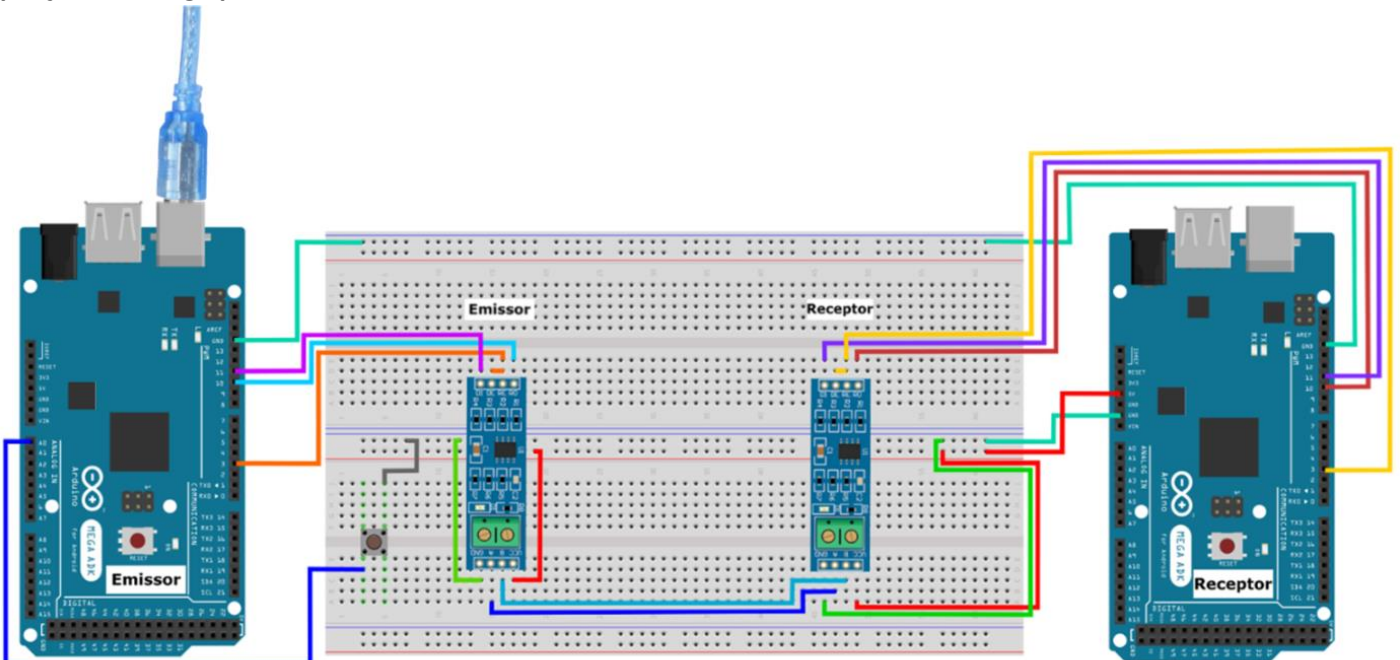
Conexão Módulo Emissor

- Pino VCC Módulo Emissor conectado ao 5V
- Pino B Módulo Emissor conectado ao B do Módulo Receptor
- Pino A Módulo Emissor conectado ao A do Módulo Receptor
- Pino GND Módulo Emissor conectado ao GND
- Pino RO Módulo Emissor conectado ao Pino 10 Arduino Emissor
- Pino RE e DE do Módulo Emissor conectados ao Pino 3 Arduino Emissor
- Pino DI Módulo Emissor conectados ao Pino 11 Arduino Emissor
- Pino A0 Arduino Emissor conectado ao pino 1 do botão, o pino 2 do botão no GND

Conexão Módulo Receptor

- Pino VCC Módulo Receptor conectado ao 5V
- Pino B Módulo Receptor conectado ao B do Módulo Emissor
- Pino A Módulo Receptor conectado ao A do Módulo Emissor
- Pino GND Módulo Receptor conectado ao GND
- Pino RO Módulo Receptor conectado ao Pino 10 Arduino Receptor
- Pino RE e DE Módulo Receptor conectados ao Pino 3 Arduino Receptor
- Pino DI Módulo Receptor conectados ao Pino 11 Arduino Receptor
- Pino GND Arduino Receptor conectado ao GND Arduino Emissor

Compilação do código para o Emissor



Para utilização do Módulo conversor TTL para RS485 Arduino como Emissor, conecte o cabo de comunicação do Arduino Emissor com o computador, e em seguida copie o software “Emissor” disponibilizado na descrição do vídeo ou em “Roteiro de Teste de funcionamento Módulo conversor TTL para RS485”; cole o software na IDE do Arduino e compile-o para o Arduino Emissor, como na imagem a seguir.

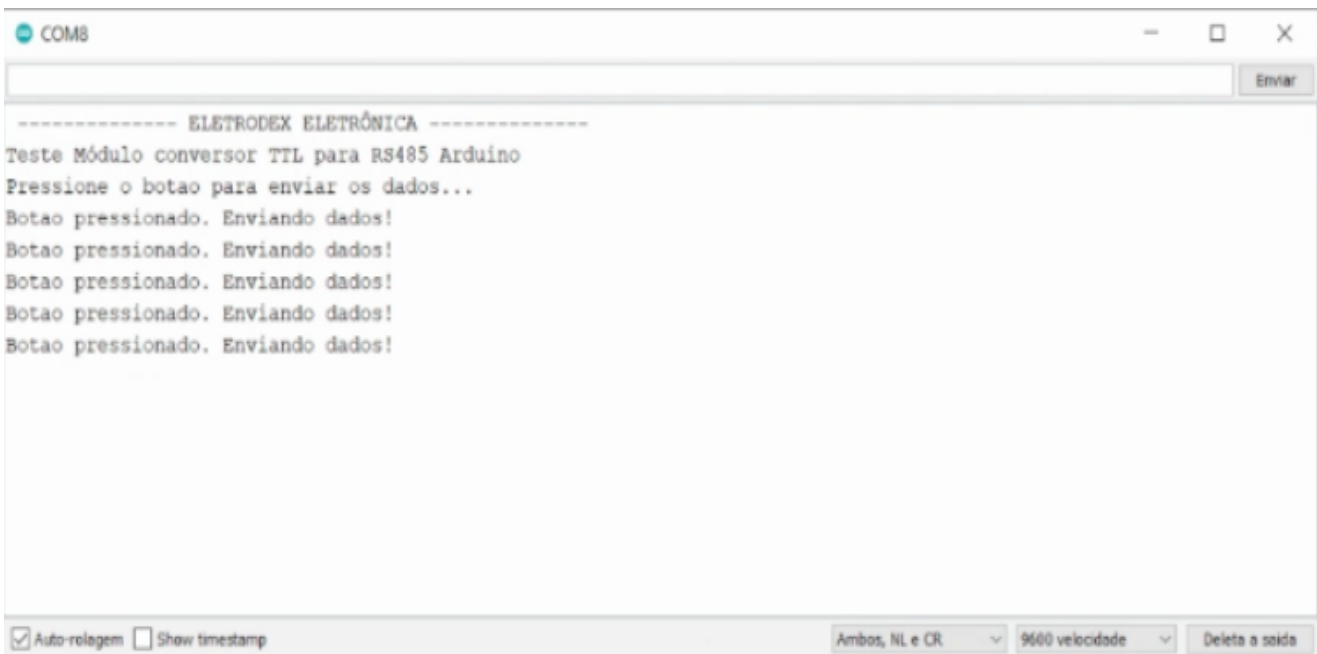
```

sketch_dec09a | Arduino 1.8.19 (Windows Store 1.8.57.0)
Arquivo Editar Sketch Ferramentas Ajuda

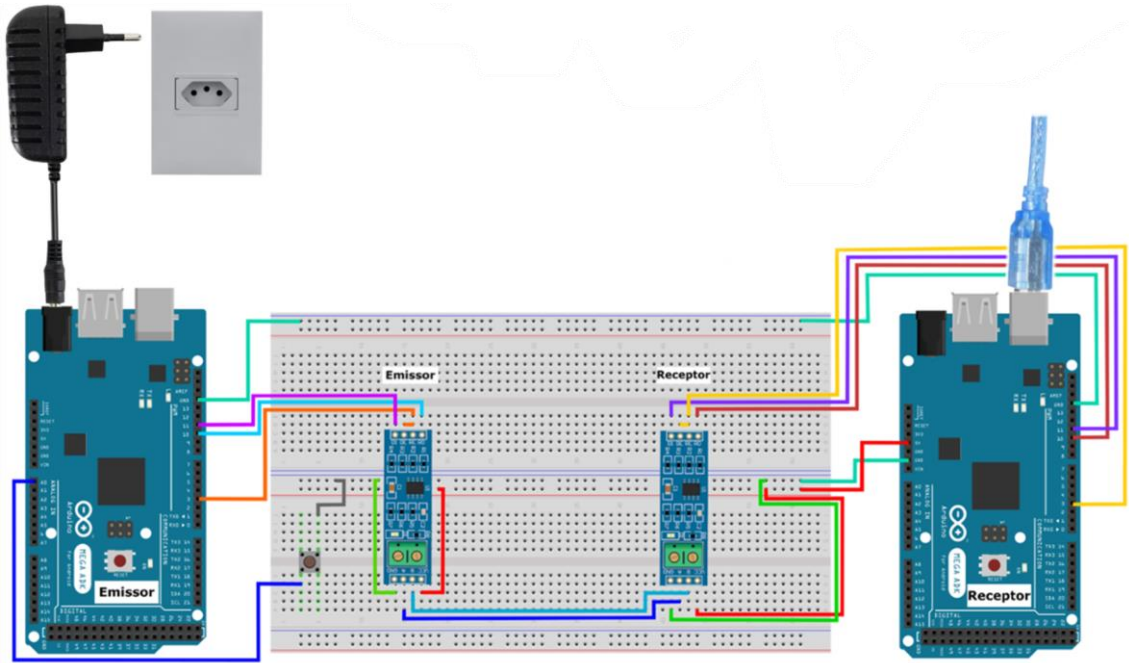
sketch_dec09a
1 | ***** Eletrodex Eletrônica *****
2 |     Teste Conversor TTL para RS485 Arduino - EMISSOR
3 |     *****/
4 |
5 | #include <SoftwareSerial.h>
6 |
7 | //Pinos de comunicacao serial do modulo RS485
8 | #define Pino_RS485_RX    10
9 | #define Pino_RS485_TX    11
10 |
11 | //Pino de controle Emissao/recepcao
12 | #define SerialTxControle 3
13 |
14 | #define RS485Emit    HIGH
15 | #define RS485Receber    LOW
16 |
17 | //Define led 13 para mostrar atividade na comunicacao
18 | #define Pin13LED    13
19 |
20 | //Cria a serial por software para conexao com modulo RS485
21 | SoftwareSerial RS485Serial(Pino_RS485_RX, Pino_RS485_TX);
22 |
23 | void setup()
24 | {
25 |     //Inicializa a serial do Arduino
26 |     Serial.begin(9600);
27 |     Serial.println("Modulo Emissor");
28 |     Serial.println("Pressione o botao para enviar os dados...");

```

Abra o monitor Serial, e pressione o Botão para enviar os dados ao receptor. A mensagem “Botão pressionado. Enviando dados!” será exibida.



Compilação do código para o Receptor



Os dados de botão pressionado serão enviados ao receptor, porém, como ainda não compilamos o código para ele, nada será recebido. Para recebimento, iremos compilar o código “RECEPTOR” para o Arduino receptor, de modo que ele possa receber os dados por meio do Módulo Receptor conversor TTL para RS485. Para isso, vamos alimentar o Arduino emissor com uma fonte externa. E colocaremos o cabo de interface entre o Arduino Receptor e o computador, para a transferência do software para o receptor, e análise do recebimento do sinal emitido pelo Botão.

```

sketch_dec09a | Arduino 1.8.19 (Windows Store 1.8.57.0)
Arquivo Editar Sketch Ferramentas Ajuda
sketch_dec09a $
1 /***** Eletrodex Eletrônica *****/
2     Teste Conversor TTL para RS485 Arduino - RECEPTOR
3 *****/
4
5 #include <SoftwareSerial.h>
6
7 //Pinos de comunicacao serial do modulo RS485
8 #define Pino_RS485_RX    10
9 #define Pino_RS485_TX    11
10
11 //Pino de controle transmissao/recepcao
12 #define SerialTxControle 3
13
14 #define RS485Transmit    HIGH
15 #define RS485Receber    LOW
16
17 //Define led 13 para mostrar atividade na comunicacao
18 #define Pin13LED        13
19
20 //Cria a serial por software para conexao com modulo RS485
21 SoftwareSerial RS485Serial(Pino_RS485_RX, Pino_RS485_TX);
22
23 //Armazena os dados que chegam pela serial
24 String inputString = "";
25 //Variavel de string completa
26 boolean stringComplete = false;
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) em COM8

```

Abra o monitor Serial do programa compilado para o Receptor, e pressione o Botão conectado ao emissor para enviar os dados ao receptor. A mensagem “Botão pressionado. Recebendo Dados” será exibida, comprovando a comunicação entre Emissor e Receptor, com atuação dos Módulos conversores TTL para RS485.



Softwares RECEPTOR e EMISSOR

Software RECEPTOR:

```

/***** Eletronex Eletrônica *****/
  Teste Conversor TTL para RS485 Arduino - RECEPTOR
  *****/

#include <SoftwareSerial.h>

//Pinos de comunicacao serial do modulo RS485
#define Pino_RS485_RX 10
#define Pino_RS485_TX 11

//Pino de controle transmissao/recepcao
#define SerialTxControle 3

#define RS485Emissor HIGH
#define RS485Receptor LOW

//Define led 13 para mostrar atividade na comunicacao
#define Pin13LED 13

//Cria a serial por software para conexao com modulo RS485
SoftwareSerial RS485Serial(Pino_RS485_RX, Pino_RS485_TX);

//Armazena os dados que chegam pela serial
String inputString = "";
//Variavel de string completa
boolean stringComplete = false;

void setup()
{
  //Inicializa a serial do Arduino
  Serial.begin(9600);
  Serial.println("----- ELETRONEX ELETRÔNICA -----");
  Serial.println("Teste Módulo conversor TTL para RS485 Arduino");
  Serial.println("Aguardando dados do Emissor...");
  Serial.println("Pressione o Botão...");

  pinMode(Pin13LED, OUTPUT);

```

```

pinMode(SerialTxControle, OUTPUT);

//Coloca o modulo RS485 em modo de recepcao
digitalWrite(SerialTxControle, RS485Receptor);

//Inicializa a serial do modulo RS485
RS485Serial.begin(4800);
}

void loop()
{
//Recebe os dados do RS485 via porta serial
if (RS485Serial.available())
{
while (RS485Serial.available())
{
//Recebe os dados e monta a string
char inChar = (char)RS485Serial.read();
inputString += inChar;

if (inChar == '\n')
{
//Mostra no Serial Monitor a string recebida
Serial.print(inputString);
stringComplete = true;
inputString = "";
}
}
}
}
}

```

Software EMISSOR:

```

/***** EletróDEX Eletrônica *****/
  Teste Conversor TTL para RS485 Arduino - EMISSOR
*****/

#include <SoftwareSerial.h>

//Pinos de comunicacao serial do modulo RS485
#define Pino_RS485_RX 10
#define Pino_RS485_TX 11

//Pino de controle Emissao/recepcao
#define SerialTxControle 3

#define RS485Emit HIGH
#define RS485Receber LOW

//Define led 13 para mostrar atividade na comunicacao
#define Pin13LED 13

//Cria a serial por software para conexao com modulo RS485
SoftwareSerial RS485Serial(Pino_RS485_RX, Pino_RS485_TX);

void setup()
{
//Inicializa a serial do Arduino
Serial.begin(9600);
Serial.println("----- ELETRÓDEX ELETRÔNICA -----");
Serial.println("Teste Módulo conversor TTL para RS485 Arduino");
Serial.println("Pressione o botao para enviar os dados...");

pinMode(Pin13LED, OUTPUT);
pinMode(SerialTxControle, OUTPUT);

```

```
//Inicializa a serial do modulo RS485
RS485Serial.begin(4800);

//Seta o pino A0 como entrada e habilita o pull up
pinMode(A0, INPUT_PULLUP);
}

void loop()
{
  //Verifica se o botao foi pressionado
  int valor = digitalRead(A0);
  if (valor == 0)
  {
    Serial.println("Botao pressionado. Enviando dados!");

    //Habilita o modulo para transmissao
    digitalWrite(SerialTxControle, RS485Emit);

    //Envia a string
    RS485Serial.println("Botao pressionado. Recebendo Dados");

    //Liga o led 13 para mostrar que ha conexao
    digitalWrite(Pin13LED, HIGH);
    delay(10);
    digitalWrite(Pin13LED, LOW);

    //Desabilita o modulo para transmissao
    digitalWrite(SerialTxControle, RS485Receber);
    while (digitalRead(A0) == 0)
    {
      delay(50);
    }
  }
}
```