

## Softwares utilizados para Teste de Funcionamento do Módulo conversor TTL para RS485 Arduino como RECEPTOR e EMISSOR

### Software RECEPTOR:

```
/****** Eletrodex 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(" ----- ELETRODEX 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:

```
/****** Eletrodex 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("----- ELETRODEX 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);  
    }  
  }  
}
```