



COMUNICAÇÃO DE DADOS – ATIVIDADE 1

ABNER ROCHA
JARBAS REZENDE
LUAN RIBEIRO

SERRA
12 - 2017

ABNER ROCHA
JARBAS REZENDE
LUAN RIBEIRO

COMUNICAÇÃO DE DADOS – ATIVIDADE 1

Relatório apresentado ao Professor Rafael Emerick, como atividade avaliativa, da disciplina de Comunicação de dados da turma do 9º período do curso de Engenharia de Controle e Automação.

Instituto Federal do Espírito Santo
Serra - 12/2017

LISTA DE ILUSTRAÇÕES

Figura 1 - Interligação de dispositivos em half duplex para RS-485.	6
Figura 2 - Esquema dos dispositivos interligados.	7
Figura 3 - Recursos necessários para desenvolvimento das placas.	8
Figura 4 - Planilha de custos.	9
Figura 5 - Diagrama do linkit Smart 7688 Duo.	10
Figura 6 - Diagrama do nodeMCU V3 ESP8266.	11
Figura 7 - Esquema de ligação do MAX485.	11
Figura 8 - Esquema de ligação para o acionamento da carga.	12
Figura 9 - Montagem das placas.	12

SUMÁRIO

OBJETIVO	5
INTRODUÇÃO.....	5
TOPOLOGIA	5
RS-485	6
ESQUEMA DE MONTAGEM.....	7
RECURSOS NECESSÁRIOS.....	8
PLANILHA DE CUSTOS	9
FUNCIAMENTO E DIAGRAMAS DOS CIRCUITOS	10
LAYOUT DA PLACA.....	12
REPOSITÓRIO	13
CONCLUSÃO.....	13

OBJETIVO

Aplicação dos conhecimentos desenvolvidos em aula na implementação de uma comunicação serial com o protocolo RS-485 para um mestre e dois escravos.

INTRODUÇÃO

Neste relatório será demonstrado o desenvolvimento de uma topologia de comunicação mestre-escravo utilizando o padrão de comunicação RS-485. Serão utilizados um mestre e dois escravos interligados através de um barramento com padrão de comunicação RS-485 entre eles enquanto o mestre se comunica com o computador através da USB.

Um escravo será utilizado para acender uma lâmpada através da luminosidade medida por um LDR (Resistor dependente de Luz) enquanto o outro escravo controla um ventilador através da temperatura medida por um LM35 (Sensor de temperatura).

TOPOLOGIA

A comunicação entre o mestre e os escravos se dá através de um barramento composto por par de fios trançados onde através deste par de fios os dispositivos transmitem e recebem dados. Quando requer alguma informação, o mestre aciona o escravo ao qual se deseja comunicar e o escravo ao receber a “solicitação” do mestre responde através do mesmo par de fios utilizados anteriormente, ou seja, em um mesmo instante de tempo só haverá um dispositivo utilizando o barramento, o que caracteriza a comunicação como half-duplex.

RS-485

O protocolo RS-485 foi desenvolvido para a utilização com até 32 equipamentos situados a uma distancia de até 1,2km, utilizando como meio físico, par trançado de fios e eventualmente um cabo terra.

Quando conectados em half-duplex, a interligação dos dispositivos se dá da seguinte forma:

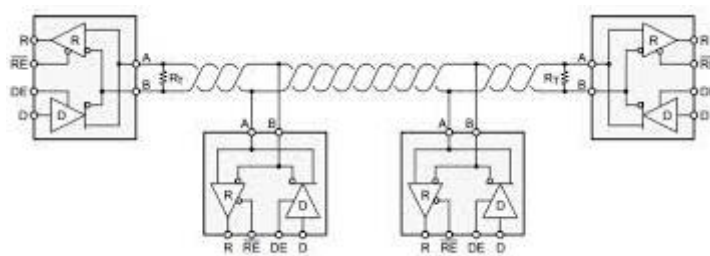


Figura 1 - Interligação de dispositivos em half duplex para RS-485.

Observe que são instalados resistores de terminação (RT) nas extremidades do barramento para evitar a reflexão de sinal e eventuais falhas de comunicação devido a esta reflexão.

O endereçamento da comunicação mestre escravo é realizado por software, uma vez que apenas um transmissor pode acessar o barramento por vez.

Como exemplo de endereçamento, no caso deste projeto, será utilizado conforme segue:

- O primeiro caractere deve ser o endereço do escravo que se deseja ler a informação:

Neste caso é possível 256 endereços, exceto o caractere “a”, usado para controle. Portanto são possíveis 255 endereços:

Definimos o endereço ‘0’ para o slave_zero e 1 para o slave_one.

- O segundo caractere deve ser o caractere de controle ‘a’ (Adress).
‘0a’ para falar com o slave_zero.
‘1a’ para falar com o slave_um.
- O terceiro caractere deve ser ‘r’ para a operação de leitura ou o valor para setup (‘0’ ou ‘1’)
‘0ar’ para ler o sensor da placa slave_zero.

'1ar' para ler o sensor da placa slave_one.

- Um quarto caractere é necessário para o caso de escrita ('s').

'0a0s' para desligar o relê da placa slave_zero.

'1a1s' para ligar o relê da placa slave_one.

Importante ressaltar que o RS-485 permite a comunicação full duplex, quando utilizados dois pares trançados.

ESQUEMA DE MONTAGEM

O esquema de montagem do sistema se dará da seguinte forma:

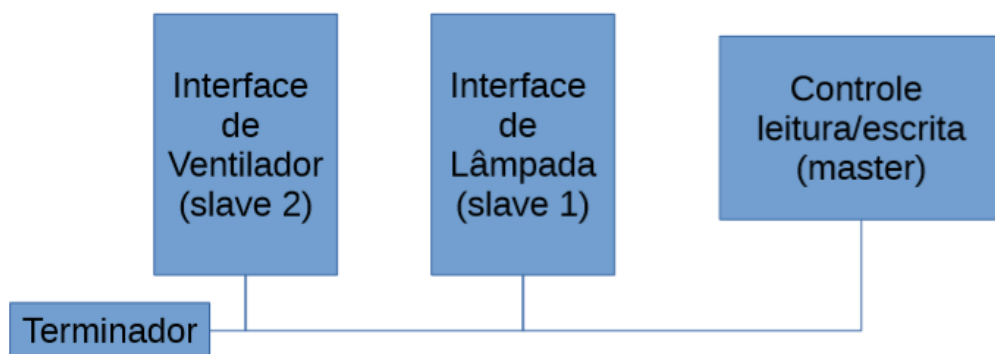


Figura 2 - Esquema dos dispositivos interligados.

RECURSOS NECESSÁRIOS

Os recursos necessários para a montagem e implementação deste projeto seguem separados para cada dispositivo:





























MESTRE		Escravo 1		Escravo 2	
Linkit7698		NODEMCU ESP8266		NODEMCU ESP8266	
Placa perfurada de circuito impresso		Placa perfurada de circuito impresso		Placa perfurada de circuito impresso	
Resistores		Resistores		Resistores	
Conversor Bidirecional 3.3V-5V		Conversor Bidirecional 3.3V-5V		Conversor Bidirecional 3.3V-5V	
CI MAX485		CI MAX485		CI MAX485	
Borne de 2 terminais		Borne de 2 terminais		Borne de 2 terminais	
		LDR		LM35	
		Transistor		Transistor	
		Relé		Relé	
		Tomada padrão (Plug)		Tomada padrão (Plug)	
		Fonte de alimentação		Fonte de alimentação	

Figura 3 - Recursos necessários para desenvolvimento das placas.

Além dos recursos listados ainda serão necessários os pares trançados como meio de transmissão e um computador para comunicação com o mestre.

PLANILHA DE CUSTOS

O custo total do projeto segue descrito na tabela abaixo:

Master			
Descrição:	Valor	Quantidade	Total
Linkit Smart 7688 duo	R\$ 160,00	1	R\$ 160,00
Placa Perfurada	R\$ 5,20	1	R\$ 5,20
Resistores	R\$ 0,15	5	R\$ 0,75
Conversos Bidirecional 3.3V-5V	R\$ 7,57	1	R\$ 7,57
MAX485	R\$ 3,20	1	R\$ 3,20
Borne 2 polos	R\$ 0,68	1	R\$ 0,68
TOTAL	R\$ 177,40		
Slave1			
Descrição:	Valor	Quantidade	Total
NodeMCU	R\$ 30,00	1	R\$ 30,00
Placa Perfurada	R\$ 5,20	1	R\$ 5,20
Placa Perfurada	R\$ 5,20	1	R\$ 5,20
Resistores	R\$ 0,15	5	R\$ 0,75
Conversos Bidirecional 3.3V-5V	R\$ 7,57	1	R\$ 7,57
MAX485	R\$ 3,20	1	R\$ 3,20
Borne 2 polos	R\$ 0,68	2	R\$ 1,36
Transistor	R\$ 2,13	1	R\$ 2,13
LDR	R\$ 1,36	1	R\$ 1,36
Relé 5V - 125Vca - 10A	R\$ 2,47	1	R\$ 2,47
Plug (Tomada)	R\$ 2,49	1	R\$ 2,49
Fonte de alimentação	R\$ 12,00	1	R\$ 12,00
TOTAL	R\$ 73,73		
Slave2			
Descrição:	Valor	Quantidade	Total
NodeMCU	R\$ 30,00	1	R\$ 30,00
Placa Perfurada	R\$ 5,20	1	R\$ 5,20
Placa Perfurada	R\$ 5,20	1	R\$ 5,20
Resistores	R\$ 0,15	5	R\$ 0,75
Conversos Bidirecional 3.3V-5V	R\$ 7,57	1	R\$ 7,57
MAX485	R\$ 3,20	1	R\$ 3,20
Borne 2 polos	R\$ 0,68	2	R\$ 1,36
Transistor	R\$ 2,13	1	R\$ 2,13
LM35	R\$ 4,90	1	R\$ 4,90
Relé 5V - 125Vca - 10A	R\$ 2,47	1	R\$ 2,47
Plug (Tomada)	R\$ 2,49	1	R\$ 2,49
Fonte de alimentação	R\$ 12,00	1	R\$ 12,00
TOTAL	R\$ 77,27		
CUSTO TOTAL	R\$ 328,40		

Figura 4 - Planilha de custos.

FUNCIAMENTO E DIAGRAMAS DOS CIRCUITOS

Para a montagem do circuito do mestre foi utilizado o linkit 7688 duo, placa de desenvolvimento compatível com o arduino e baseada na distribuição Linux OpenWrt, com microprocessador MT7688 e microcontrolador ATmega32U4.

O MT7688 é um processador MIPS com WiFi integrado e conta também com Ethernet, USB e interface para cartão microSD, sendo assim, o linkit é capaz de rodar um sistema operacional Linux.

Características da linkit:

- Processador MT7688 MIPS 580 MHz
- Conectividade Wi-Fi 802.11 b/g/n (2.4G)
- Sinasi GPIO, I2C, SPI, UART, PWM e Ethernet (Ethernet precisa de adaptador!)
- 128MB de memória RAM DDR2
- 32MB de memória Flash
- Interface USB Host (permite conectar Webcam, mouse, teclado, pendrive, etc).
- Slot para cartão microSD
- Arduino – ATmega32U4

Segue imagem indicando a pinagem da linkit:

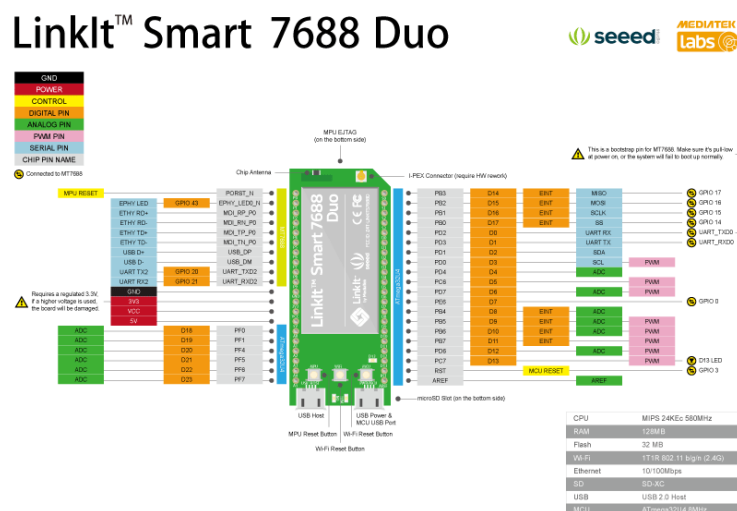


Figura 5 - Diagrama do linkit Smart 7688 Duo.

Para o desenvolvimento dos escravos a placa utilizada foi o Nodemcu V3 ESP8266, conforme sugerido no escopo da atividade.

O nodemcu é uma placa de desenvolvimento que combina o chip ESP8266, uma interface usb-serial e um regulador de tensão de 3.3V. A programação do placa é feita utilizando LUA ou a própria IDE do arduino, usando a comunicação via cabo micro-usb ou WiFi.

Abaixo segue a imagem descrevendo a pinagem da placa:

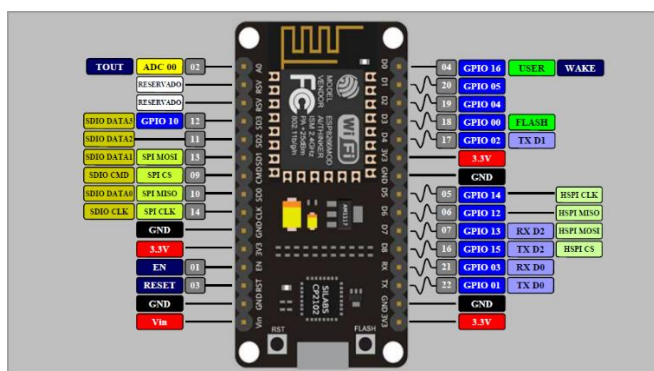


Figura 6 - Diagrama do nodeMCU V3 ESP8266

Para a comunicação RS-485 utilizamos um conversor de tensão de 3,3V para 5V em todas as placas, com a finalidade de elevar o nível de tensão do sinal para 5V, padrão adotado no RS-485.

O sinal de saída do conversor foi ligado no CI MAX485 que converte o sinal de entrada para o padrão de comunicação RS-485. O esquema de ligação do MAX485 segue abaixo:

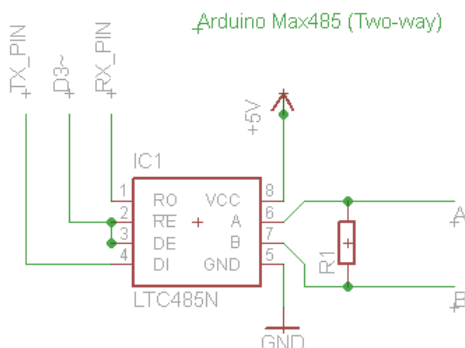


Figura 7 - Esquema de ligação do MAX485.

Os pinos 2 e 3 do max485 definem se o CI irá estar em modo de transmissão ou recebimento.

Para o acionamento da carga foi necessário a utilização de um transistor, uma vez que a saída do nodeMCU não é suficiente para tal, sendo assim, a saída do circuito aciona o transistor e este, por sua vez, aciona o relé aonde está conectada a carga, conforme segue no diagrama:

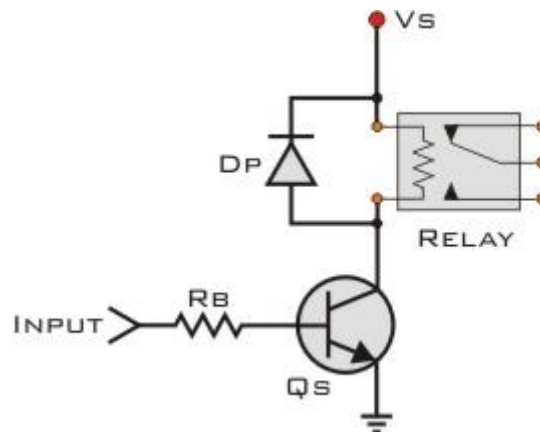


Figura 8 - Esquema de ligação para o acionamento da carga.

LAYOUT DA PLACA

Para a montagem dos circuitos foram utilizadas placas perfurada tendo como resultado final a imagem que segue:



Figura 9 - Montagem das placas

REPOSITÓRIO

Os arquivos utilizados no desenvolvimento do projeto estão armazenados no repositório git que segue:

<https://github.com/ssribeiro/abjalu.git>

No repositório está o código desenvolvido, a planilha de custos, o relatório e alguns arquivos que auxiliaram no desenvolvimento do projeto.

CONCLUSÃO

Ao fim do desenvolvimento deste projeto foi possível observar a comunicação RS-485 funcionando conforme a teoria citada em aula.

Além disso foram observadas dificuldades, principalmente no desenvolvimento do hardware, uma vez que, diversas vezes tivemos problemas com mau-contato, fator que interfere diretamente no resultado final do projeto. A utilização de placa perfurada para o desenvolvimento do projeto também não foi considerada uma boa escolha, devido às dificuldades encontradas na montagem propriamente dita.

Podemos verificar a comunicação RS-485 funcionando dentro do padrão estabelecido e sem perdas de pacotes.

