

INSTITUTO FEDERAL DO ESPÍRITO SANTO
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

DIEGO DE DEUS
GABRIEL SCHERR
JORGE

RELATÓRIO DE COMUNICAÇÃO DE DADOS

SERRA
2017

DIEGO DE DEUS
GABRIEL SCHERR MARTINS
JORGE

RELATÓRIO DE COMUNICAÇÃO DE DADOS

Trabalho apresentado à Coordenadoria do Curso de Engenharia de Controle e Automação do Instituto Federal do Espírito Santo como requisito parcial de avaliação.

Orientador: Prof. Mr. Rafael Emerick.

SERRA

2017

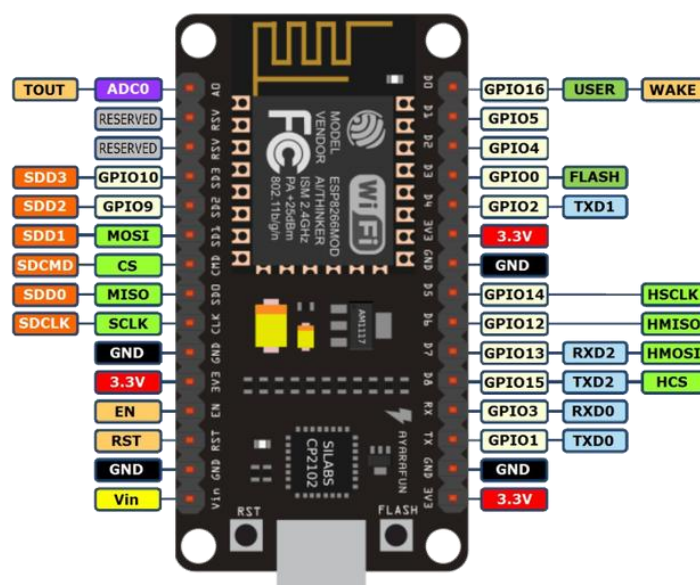
INTRODUÇÃO E MOTIVAÇÃO

O intuito deste trabalho é criar uma topologia de comunicação Mestre-escravo, e implementação de comunicação sensoriada utilizando o padrão RS485.

1. NodeMCU

O micro controlador escolhido neste trabalho para o controle do sensoriamento que disponibiliza um servidor Web HTTP, com páginas que disponibilizem HTML foi o NodeMCU, que é uma plataforma *open source* com a possibilidade de utilizar a linguagem de script Lua, uma linguagem de múltiplos paradigmas, oferecendo um pequeno conjunto de características gerais que podem ser estendidas para encaixar diferentes tipos de problemas, em vez de fornecer uma especificação mais complexa e rígida para combinar com um único paradigma, essa linguagem permite que programadores quando implementam nomes, classes, e outras funções, empreguem poderosas técnicas de programação funcional e completos escopos lexicais.

O NodeMCU contém o módulo ESP8266 para conexão via Wi-Fi, usa uma interface usb-serial e um regulador de tensão 3.3V, utilizando a comunicação via cabo micro-usb e conta com 11 pinos de I/O e um conversor analógico-digital.

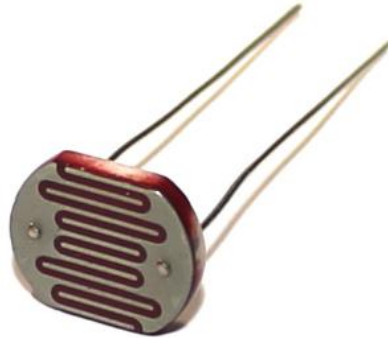


Especificações:

- ESP8266 ESP-12F
- Wireless padrão 802.11 b/g/n
- Antena embutida
- Conector micro-usb
- Modos de operação: STA/AP/STA+AP
- Suporta 5 conexões TCP/IP
- Portas GPIO: 11
- GPIO com funções de PWM, I2C, SPI, etc
- Tensão de operação: 4,5 ~ 9V
- Taxa de transferência: 110-460800bps
- Suporta Upgrade remoto de firmware
- Conversor analógico digital (ADC)
- Distância entre pinos: 2,54mm
- Dimensões: 49 x 25,5 x 7 mm

2.2 SENSOR DE LUMINOSIDADE

O sensor usado para medir o nível de luminosidade, foi um LDR que apresenta as características elétricas mostrada na tabela abaixo:



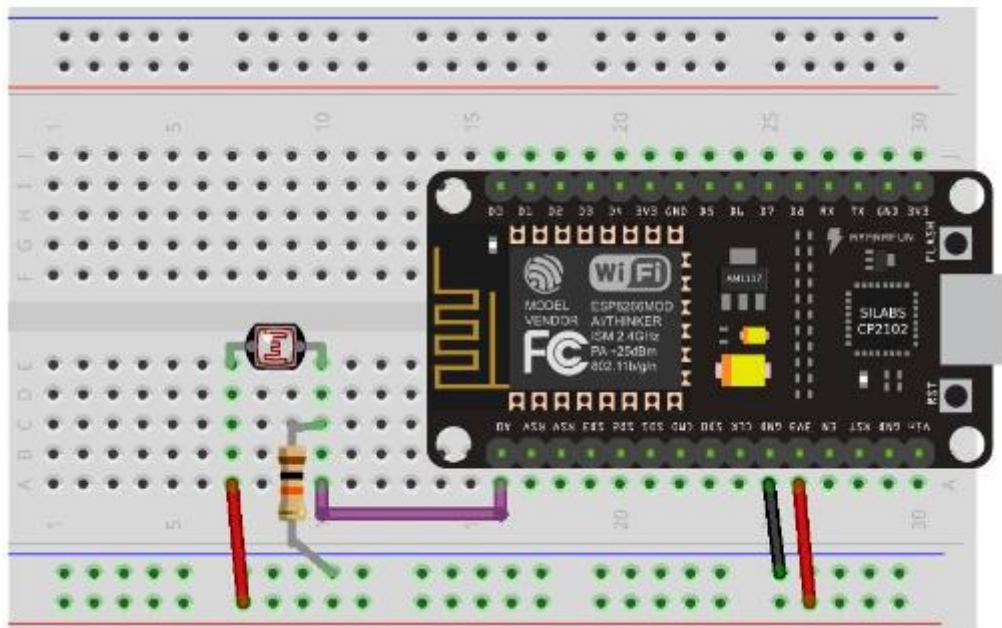
Electrical Characteristics

Parameter	Conditions	Min	Typ	Max	Unit
Cell resistance	1000 LUX	-	400	-	Ohm
	10 LUX	-	9	-	K Ohm
Dark Resistance	-	-	1	-	M Ohm
Dark Capacitance	-	-	3.5	-	pF
Rise Time	1000 LUX	-	2.8	-	ms
	10 LUX	-	18	-	ms
Fall Time	1000 LUX	-	48	-	ms
	10 LUX	-	120	-	ms
Voltage AC/DC Peak		-	-	320	V max
Current		-	-	75	mA max
Power Dissipation				100	mW max
Operating Temperature		-60	-	+75	Deg. C

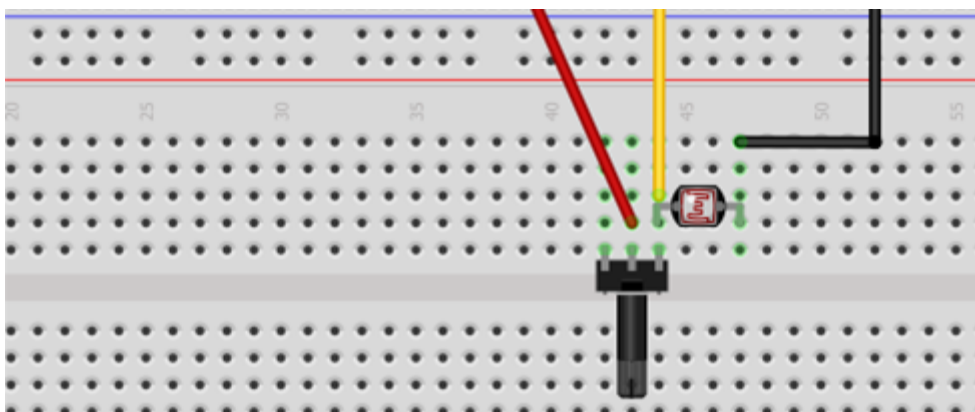
Para fazer a conexão do sensor de luminosidade (LDR) no módulo NodeMCU foram utilizados os seguintes componentes:

- NodeMCU
- LDR / fotoresistor
- Resistor de 10k ohm
- Proto Board
- Computador com Arduino IDE com biblioteca ESP8266 instalada
- Cabo USB
- Jumpers

O LDR possui uma saída analógica por isso é necessário conectá-lo no pino A0 do módulo.



Foram realizados testes de luminosidade em um laboratório de pesquisa e percebemos que o resistor de 10 K ohm se adequou ao fator de iluminação da sala. Mas se for preciso fazer um teste ou a implementação desse sensor em um outro lugar, foi colocado um potenciômetro com o valor de 10 K ohm que apresenta uma variação de resistência que pode se adequar com o fator de luminosidade do LDR.



2. COMUNICAÇÃO

O escravo tem um sensoriamento de leitura(sensor) e um acionamento de escrita (atuador). O mestre é ligado a um computador, de desse serão disparados os comandos de leitura/escrita do respectivo escravo.

3. TABELA DE GASTOS

Os gastos para a construção do protótipo, com referência para o mês de agosto de 2017 foi de R\$160,50, o que é considerado um valor baixo visto que foram utilizados apenas componentes simples, mas que cumpriram com a especificação do trabalho.

ORÇAMENTO	
COMPONENTES	PREÇO
03 NodeMCU	R\$ 120,00
LDR	R\$ 1,00
2 Protoboard	R\$ 14,00
Jumpers	R\$ 10,00
02 MAX	R\$15,00
Resistor de 100 ohms	R\$ 0,50
TOTAL	R\$ 160,50

4. REPOSITÓRIO

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

<https://github.com/dedeusdiego/ComDadosTime3>

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

5. CONCLUSÃO

A comunicação RS-485 funciona conforme a teoria citada em aula e em conjunto com o wifi.

Além disso, problemas como mau-contato surgiram no projeto, fator que interfere diretamente no resultado final.

Vemos que na comunicação via wifi, devido ao uso do ar como meio físico, não existem os problemas de falha de comunicação devido a problemas do meio físico como a reflexão do sinal pela falta de terminador ou até mesmo os problemas com conectores, além de ser possível eliminar a utilização dos fios.