

October 7, 2016

Procedimento de comunicação Raspberry 3-Arduino Uno

Durante a reunião na casa do Paulo na última sexta feira (30/09), eu e a Bel ficamos responsáveis por começar a realizar a comunicação. A gente montou um circuito simples em uma protoboard, seguindo o diagrama da Figura 1, descrito no link <http://blog.filipeflop.com/arduino/comunicacao-serial-arduino-com-raspberry-pi.html>. Diagrama dos pinos da raspberry <https://pinout.xyz/pinout/uart>

Obs: Necessita de resistores em série para nivelar a corrente entre os dispositivos, visto que as portas Tx e Rx do Arduino trabalham em 5.5v e as da Raspberry trabalham em 3.3v, isso poderia ser uma causa para danificar a placa Raspberry. Portanto deve-se tomar muito cuidado ao se manusear um circuito de comunicação UART por meio de serialização de resistores.

NÃO TROQUE OS FIOS DE RX E TX DA RASPBERRY

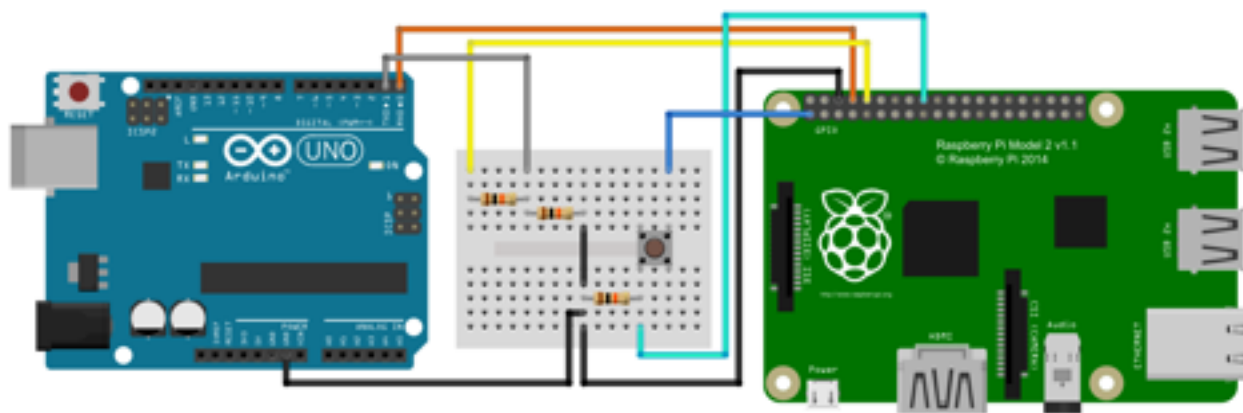


Figura 1 - Circuito para comunicação UART (Rx-Tx)

Porém esse tutorial é muito antigo e não conseguimos obter sucesso na sexta-feira. Após alguns dias estudando eu achei a causa do problema. A nova distro para o

Raspberry (Jessie) trocou algumas portas seriais de ordem e o novo modelo 3 da Raspberry, por conter um modulo bluetooth embutido, a porta /dev/ttyAMA0 (antiga porta serial Tx) foi empregada para realizar a comunicação bluetooth. Para reverter isso foi necessário, além de outras coisas, desabilitar o módulo bluetooth, trocar as portas e colocar seus links corretos. Para isso eu utilizei alguns tutoriais, na seguinte ordem:

- <https://www.abelectronics.co.uk/kb/article/1035/raspberry-pi-3-serial-port-usage>
 - Foi usado para configurar o arquivo cmdline.txt e habilitar o console serial na Rbina Jessie.
- <http://spellfoundry.com/2016/05/29/configuring-gpio-serial-port-raspbian-jessie-including-pi-3/>
 - Embasamento teórico, Configuração das portas e desabilitando o console

Após realizar esses procedimentos e tentar rodar o programa do primeiro link, não consegui obter resultados, ainda não sei o motivo por isso acontecer. Caso queira dar uma olhada nos códigos os códigos estão no nosso repositório no GitHub.

Como não consegui obter resultados eu tentei simplificar mais a aplicação e utilizei esse tutorial <https://oscarliang.com/raspberry-pi-and-arduino-connected-serial-gpio/> (Ultima parte “Connect Raspberry Pi and Arduino with a Voltage Divider ”), onde ele abre um console de comunicação, que monitora a chegada de dados na raspberry. Com isso foi visto que a comunicação está correta quanto a ser uma via de mão dupla para , porém não retornando os valores corretos ou se comportando de forma correta.

Para os próximos passos é necessário que :

- Desenvolvamos uma forma de comunicação estável, onde a gente consiga enviar e receber um dado sem sua perda.
- Após isso teremos que testar algoritmos de controle do motores e ultra-sons (bibliotecas <http://playground.arduino.cc/Code/NewPing> e <http://www.incertitudes.fr/robot/ArdumotoQuickStart.pdf> (ardumoto))
- Após testado algoritmos simples, é necessário evoluir o código de identificação da bolinha para trabalhar com o GPIO da Raspberry.
- Após o funcionamento da comunicação envolvendo a identificação da bolinha, deve-se atuar nos motores a partir de estímulos causados pelo posicionamento e distância obtidas.
- Criação de um algoritmo para detecção de obstáculos e evitá-los
- Ser feliz

Obs: As pilhas estão sem carregador, então não deve dar pra testar por muito tempo os motores, porém todo o resto funciona corretamente apenas com o cabo usb.