

5th December 2012

nRF24L01 - Testes Iniciais

Hora de falar sobre nossos módulos de rádio.

Como já postado, vamos usar o nRF24L01. Ele atende todos as nossas necessidade

- É um transreceptor (envia e recebe sinais);
- É barato;
- Existe biblioteca pronta para arduino.

Compramos os módulos no dealextreme (link [\[http://dx.com/p/nrf24l01-2-4ghz-wireless-transceiver-module-126467\]](http://dx.com/p/nrf24l01-2-4ghz-wireless-transceiver-module-126467)) a US\$ 3,70.

Na verdade existem 2 bibliotecas a [Mirf](http://playground.arduino.cc/InterfacingWithHardware/Nrf24L01) [\[http://playground.arduino.cc/InterfacingWithHardware/Nrf24L01\]](http://playground.arduino.cc/InterfacingWithHardware/Nrf24L01) e a [RF24](http://maniacbug.github.com/RF24/) [\[http://maniacbug.github.com/RF24/\]](http://maniacbug.github.com/RF24/) . Existe pouco material na internet, mas depois de uma pesquisa encontramos mais sobre a RF24, e decidimos utilizá-la.

Nossas fontes de pesquisa foram:

<http://maniacbug.github.com/RF24/> [\[http://maniacbug.github.com/RF24/\]](http://maniacbug.github.com/RF24/)

Documentação da biblioteca.

<http://maniacbug.wordpress.com/> [\[http://maniacbug.wordpress.com/\]](http://maniacbug.wordpress.com/)

Blog do desenvolvedor da biblioteca.

<http://www.bajdi.com/> [\[http://www.bajdi.com/\]](http://www.bajdi.com/)

Blog de um cara que usou a biblioteca RF24 e publicou um exemplo muito simples que nos ajudou bastante. Neste post mostraremos esse exemplo levemente modificado.

As três referencias acima foram desenvolvidas para o nRF24L01+, e não para o nRF24L01 (sem '+'), que foi o que compramos.

Devem existir diferenças, mas até agora as únicas percebidas são a quantidade de pinos (o '+' possui 8 e o 'sem +' possui 10, com um Vcc e um GND a mais) e a disposição dos pinos.

Vejam a imagem abaixo. Criamos o elemento no Fritzing com a disposição de pinos como se visto desta forma.



[\[http://4.bp.blogspot.com/-](http://4.bp.blogspot.com/-Y6qOC6X8pDU/UL6tdT3sJ_I/AAAAAAAAAvs/uqadEggTeqY/s1600/nrf24l01.png)

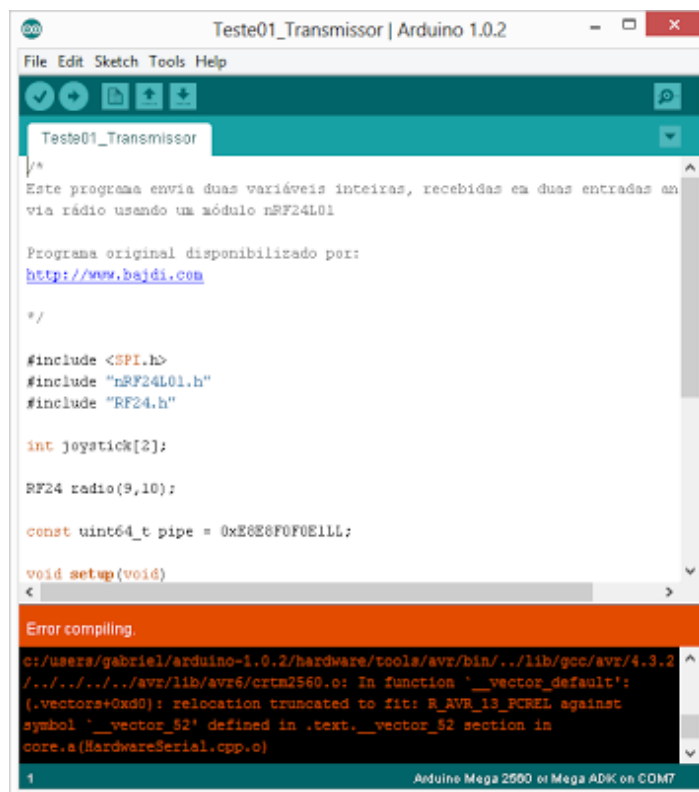
[Y6qOC6X8pDU/UL6tdT3sJ_I/AAAAAAAAAvs/uqadEggTeqY/s1600/nrf24l01.png\]](http://4.bp.blogspot.com/-Y6qOC6X8pDU/UL6tdT3sJ_I/AAAAAAAAAvs/uqadEggTeqY/s1600/nrf24l01.png)

Para usar a biblioteca RF24, bastou-nos seguir as ligações pelo nome dos pinos (e não pela disposição), os pinos duplicados Vcc e GND foram ignorados, assim como o IRQ.

No primeiro programa testado, conectamos dois potenciômetros no nosso arduino Mega e queríamos que ele transmitisse o valor captado nas entradas analógicas para um arduino Uno, onde as veríamos pelo serial monitor. Esse é o exemplo publicado pelo Badji, mencionado acima.

Por motivos desconhecidos, o código não funcionou por erro de compilação no transmissor. Depois de testar, pesquisar e tentar várias combinações, testamos outros arduino e vimos que o problema estava no arduino Mega, não no hardware, mas no software.

Não sabemos porque, mas o programa não compila se a placa selecionada no IDE do arduino for um Mega.



[<http://4.bp.blogspot.com/-4YkmURYei20/UMCmfdMOxI/AAAAAAAAAwA/cb9ld0grVOA/s1600/erro-aarduino-mega.png>]

Bem, mudamos para um Uno transmitir para um nano, e os códigos compilaram corretamente, tanto o do transmissor quanto o do receptor.

**** TRANSMISSOR ****

```
/*
Este programa envia duas variáveis inteiras, recebidas em duas entradas analógicas,
via rádio usando um módulo nRF24L01

Programa original disponibilizado por:
http://www.bajdi.com

*/

#include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"
```

```
int joystick[2];

RF24 radio(9,10);

const uint64_t pipe = 0xE8E8F0F0E1LL;

void setup(void)
{ Serial.begin(9600);
  radio.begin();
  radio.openWritingPipe(pipe);
}

void loop(void)
{
  joystick[0] = analogRead(A0);
  joystick[1] = analogRead(A1);

  radio.write( joystick, sizeof(joystick) );

}
```

**** RECEPTOR****

```
/*

Este programa recebe duas variáveis inteiras, via rádio usando um módulo nRF24L01

Programa original disponibilizado por:
http://www.bajdi.com

*/

#include <SPI.h>
#include "nRF24L01.h"
#include "RF24.h"

int joystick[2];

RF24 radio(9,10);
const uint64_t pipe = 0xE8E8F0F0E1LL;

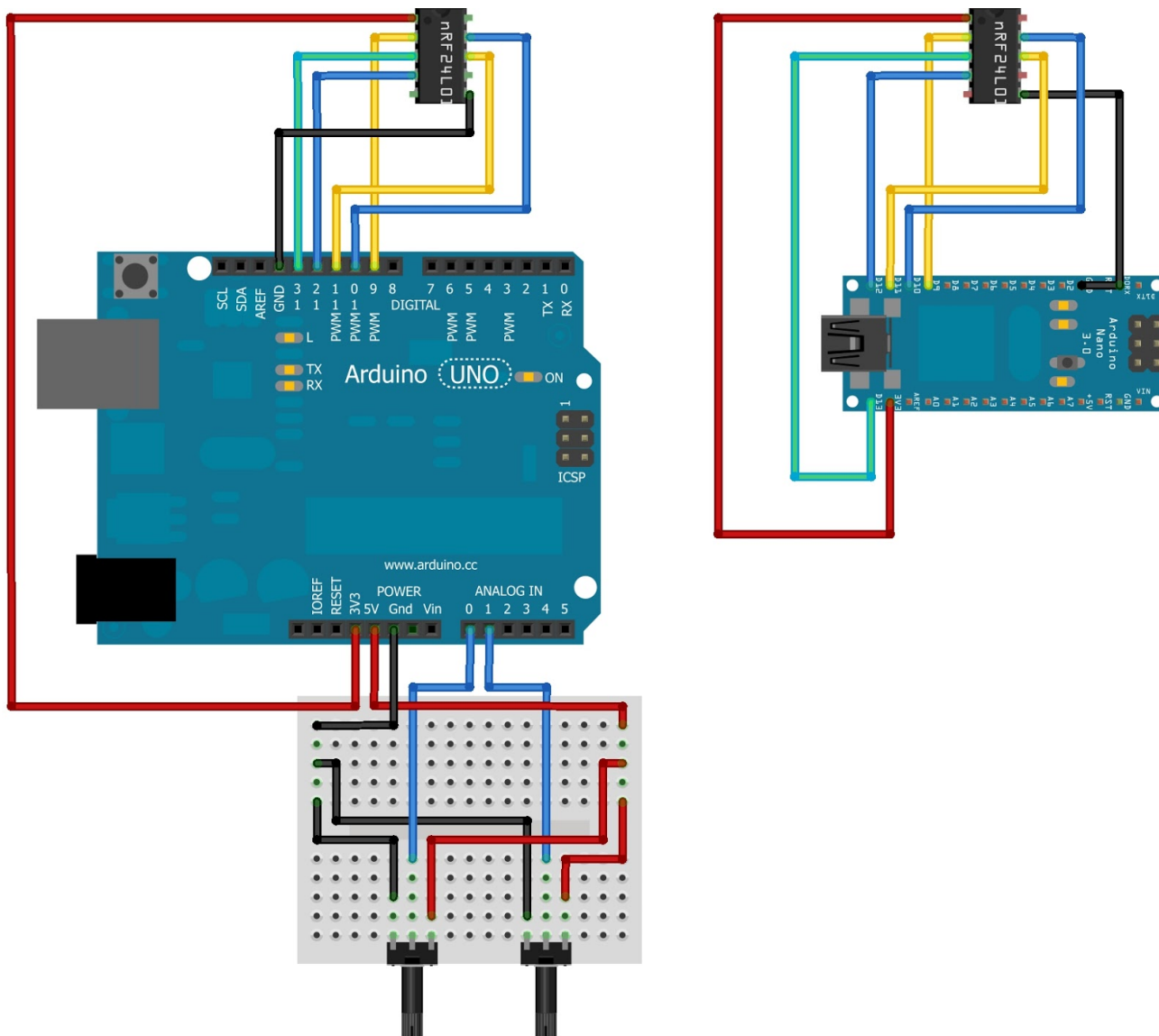
void setup(void)
{
  Serial.begin(9600);
  radio.begin();
  radio.openReadingPipe(1,pipe);
  radio.startListening();
}

void loop(void)
{
  if ( radio.available() )
  {
```

```
bool done = false;
while (!done)
{
    done = radio.read( joystick, sizeof(joystick) );
    Serial.println(joystick[0]);
    Serial.println(joystick[1]);
}
else
{
    Serial.println("Rádio não disponível");
}
// Delay para facilitar a visualização das informações no serial monitor
delay(1000);
}
```

Vamos analisar o código e a biblioteca futuramente, ou outro post.

Realizamos a ligação como o esquema abaixo:

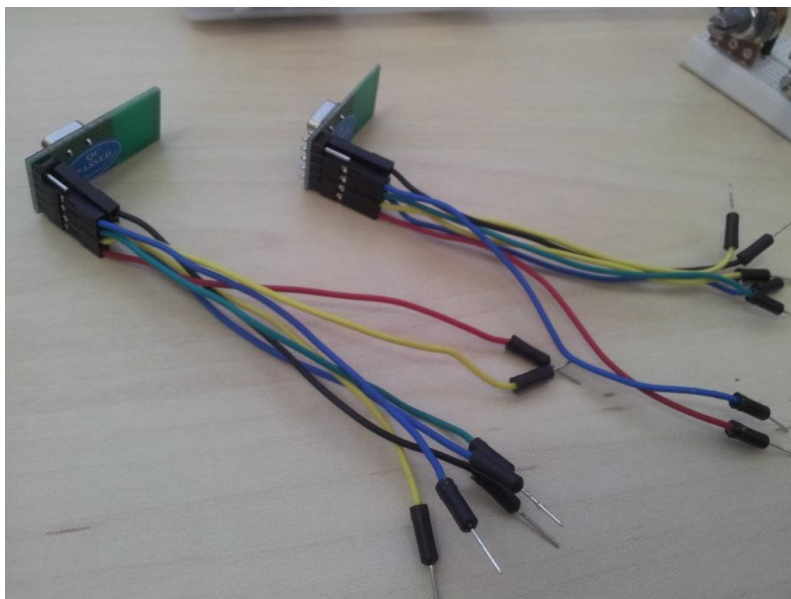
Made with  Fritzing.org

[http://2.bp.blogspot.com/-OtjuWhJnhfI/UL6m4zOzjI/AAAAAAAAAw/MsgLNttJ4rc/s1600/Teste+com+radio+01_bb.jpg]

Esquema de ligação

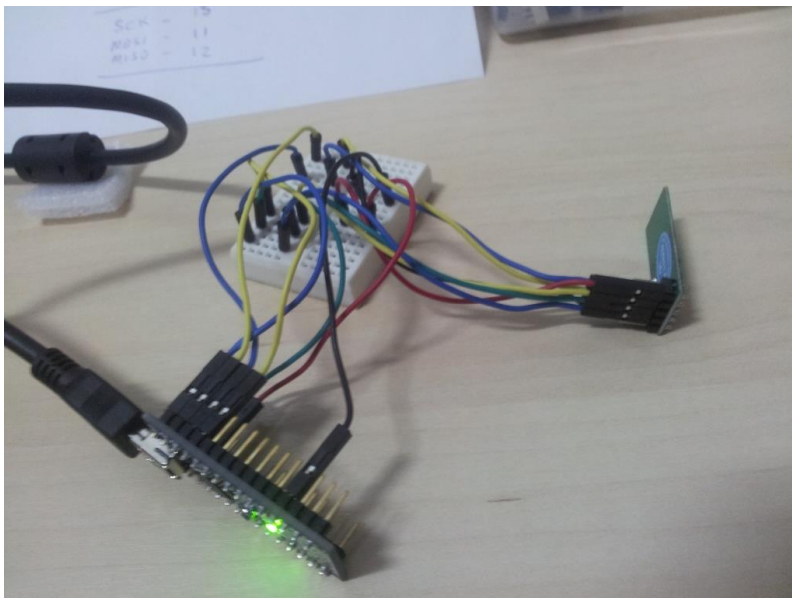
Reparem que a alimentação dos módulos é em 3.3V !

Para conectar os módulos de rádio nos arduinos, usamos jumpers fêmea/macho. No caso do Uno, foi do rádio direto para o arduino, e no ano, passamos por uma protoboard (não está desenhada no Fritzing).



[http://1.bp.blogspot.com/-1kjC4ni6ew4/UL6mzaX2YFI/AAAAAAAAAuw/WYCVux0BYAY/s1600/20121204_171419.jpg]

Módulos nRF24L01 com jumpers fêmea/macho

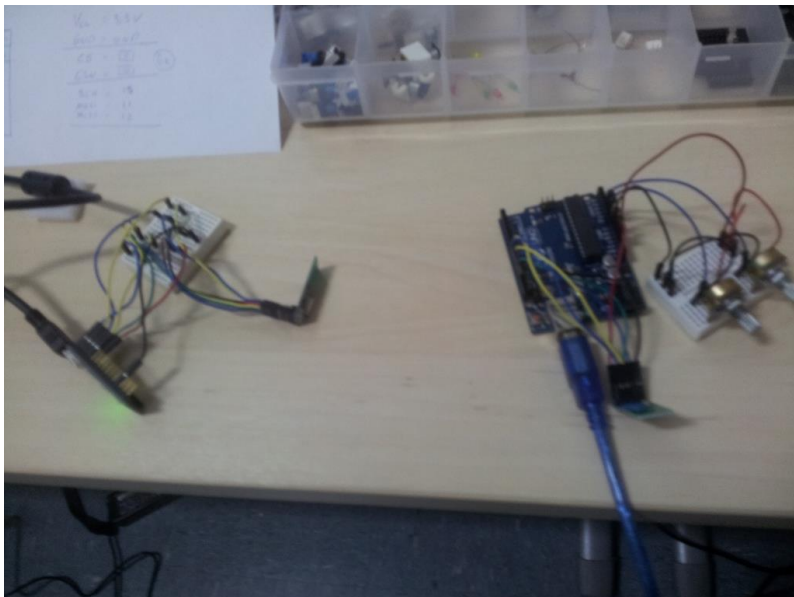


[http://3.bp.blogspot.com/-deLuJU49F1M/UL6m1RQqE_I/AAAAAAAAAvA/ouTnirBEjAo/s1600/20121204_204239.jpg]

Receptor - Arduino NANO conectado ao nRF24L01

Receptor - Arduino NANO conectado ao nRF24L01

O programa e as ligações são bem simples. A coisa funcionou na primeira tentativa.

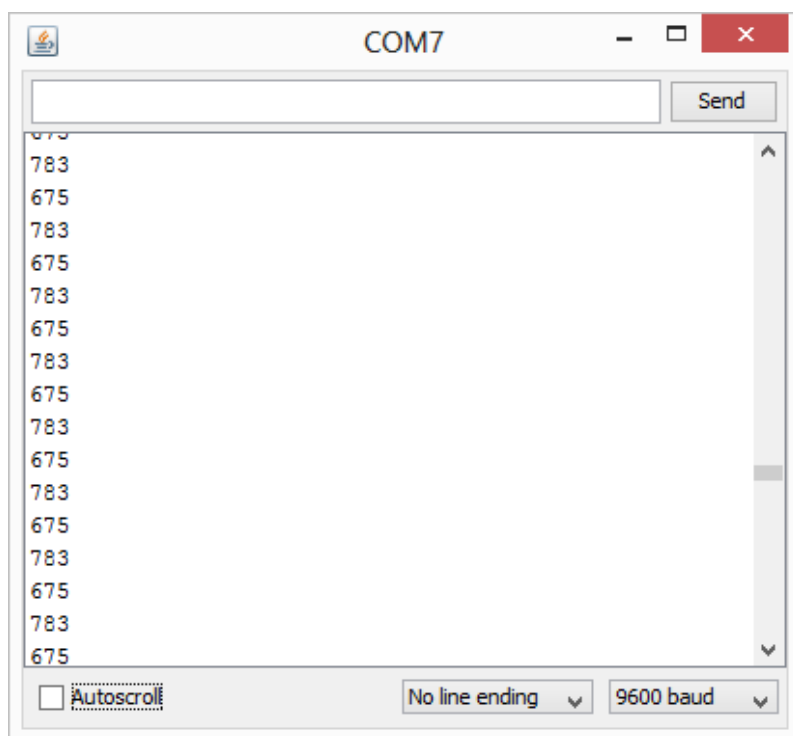


[[http://1.bp.blogspot.com/-YYADX8wk-](http://1.bp.blogspot.com/-YYADX8wk-Lw/UL6m0YXB6JI/AAAAAAAAAu4/3_p85aSAjZ0/s1600/20121204_204211.jpg)

[Lw/UL6m0YXB6JI/AAAAAAAAAu4/3_p85aSAjZ0/s1600/20121204_204211.jpg](http://1.bp.blogspot.com/-YYADX8wk-Lw/UL6m0YXB6JI/AAAAAAAAAu4/3_p85aSAjZ0/s1600/20121204_204211.jpg)]

Transmissor e Receptor

Para verificar os valores das variáveis recebidas, usamos o 'serial monitor'.



[[http://3.bp.blogspot.com/-](http://3.bp.blogspot.com/-D3fAQyRBDSI/UL6m5gBj3OI/AAAAAAAAAvQ/lj8Yn7V5Qsw/s1600/serial-monitor-radio.png)

[D3fAQyRBDSI/UL6m5gBj3OI/AAAAAAAAAvQ/lj8Yn7V5Qsw/s1600/serial-monitor-radio.png](http://3.bp.blogspot.com/-D3fAQyRBDSI/UL6m5gBj3OI/AAAAAAAAAvQ/lj8Yn7V5Qsw/s1600/serial-monitor-radio.png)]

Recepção dos valores com sucesso

Testamos o alcance de envio dos dados no corredor da universidade, mas não conseguimos determinar o limite porque o corredor acabou, então esse não será um problema, já que durante a partida não ficaremos a mais de 10 metros dos jogadores.

Precisamos agora realizar novas experiências para verificar se poderemos utilizar o arduino mega, se vamos transmitir estados de botões (como pensado a princípio) ou os valores analógicos, e principalmente como transmitir para 3 rádios.

Posted 5th December 2012 by Gabriel C Prado

Labels: [nRF24L01](#), [Arduino](#)



Add a comment

Digite seu comentário...

Comentar como: Conta do Google

Publicar

Visualizar