

# Baliza de 70cm pelo preço de um “par de fichas PL”

---

Por Pedro Ribeiro, CT7ABP

Há uns tempos entrei no mundo dos Arduínos. Desde que terminei o curso há uns anitos atrás que não mexia em microprocessadores (microcontroladores neste caso) a tão baixo nível.

Para terem uma referência, na altura o que se usava nos projectos desta temática eram processadores 80535, uma versão do 8051 da Intel melhorada pela Siemens com a integração de diversos periféricos básicos e adicionada de diversas funcionalidades de I/O (portas de entrada e/ou saída).

Isto numa era em que ainda não se falava de PICs ou ATMELs ...

Bem, mas voltando ao assunto principal. Juntando diversos componentes em torno de um Arduíno Nano, montei recentemente uma pequena baliza operando na banda dos 70cm e consegui alguns recordes interessantes com ela, tendo em conta a potência que debita.

## Componentes:

- Arduíno Nano v3.0 – Microcontrolador ATMEGA328P de 8bit da família AVR operando a 16MHz
- RFM69CW – Módulo transceptor (TX/RX) de baixo custo, vocacionado para operação isenta de licenciamento (ISM) em alarmes e controlos remotos sem fios. Versão de ~433MHz
- Conversor de níveis lógicos 5V/3.3V, bidireccional de 4 vias
- Placa de desenvolvimento (*breadboard*) de 170 pontos
- Fonte de alimentação USB de 5V (com cabo mini-USB)

Nota: Componentes disponíveis a custos muito interessantes nos *sites* eBay e aliexpress.com

## Circuito

O RFM69CW é comandado através de uma interface *Serial Peripheral Interface Bus* (SPI), forma de comunicação entre chips desenvolvida pela Motorola e usada como um standard de facto em muitos componentes de sistemas embebidos, como mostradores de cristais líquidos (LCD) ou os cartões de memória Secure Digital (SD).

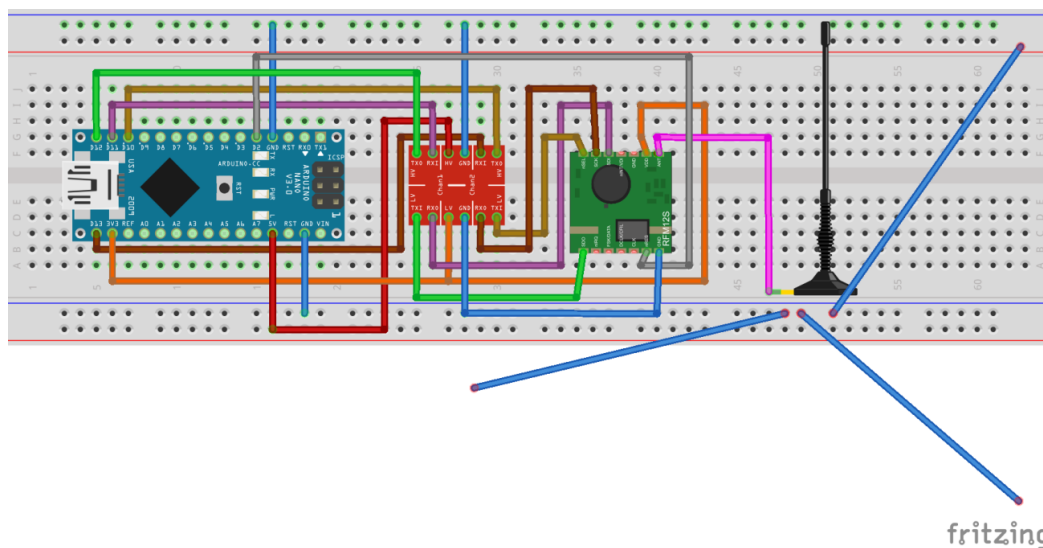
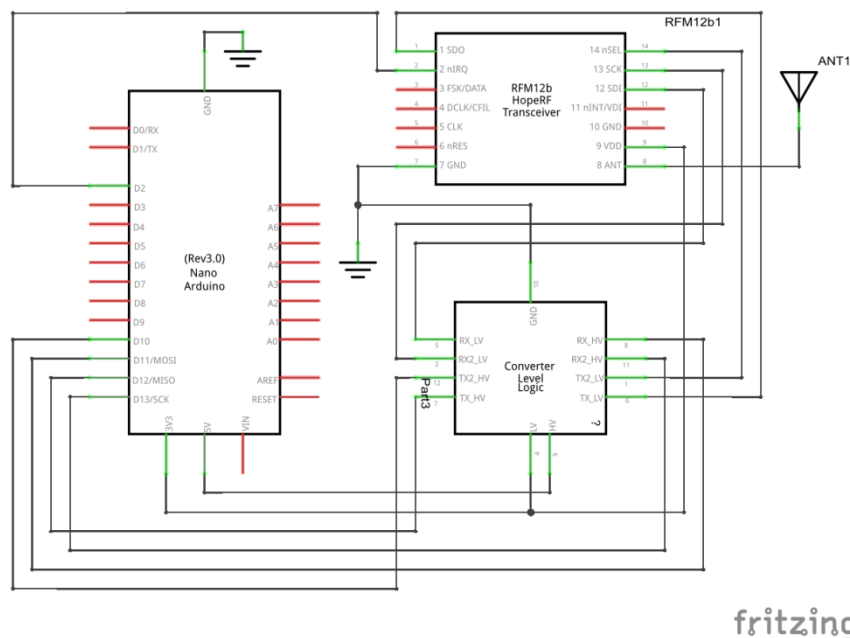
Neste módulo são usados neste projecto 5 pinos para o comando e controlo, NSS selecciona a comunicação com o módulo (podem ser usados diversos módulos SPI em paralelo no barramento, o sinal NSS (ou SS) selecciona o módulo com que o microcontrolador quer comunicar a cada momento), MOSI transporta os dados serializados do microcontrolador para os dispositivos, MISO transporta os dados para o microcontrolador e SCK (ou SCLK) dita o ritmo (*clock*) da comunicação. Finalmente, nesta aplicação é usada uma linha extra (DIO0) para sinalizar ao microcontrolador determinados eventos do módulo, como a recepção de novos dados.

Para a montagem que pretendemos é importante o facto de este módulo (RFM69CW) funcionar com uma alimentação de 3,3V, impondo também que os sinais de controlo operem com esta tensão para sinalizar o nível lógico HIGH (o 1 do binário).

Diversas versões e subversões das placas Arduíno funcionam a 3,3V, no entanto, dada a disponibilidade de uma Nano funcionando a com lógica de 5V, foi esta a usada. Teremos portanto de usar o conversor de níveis lógicos para o compatibilizar com o RFM69CW.

Como alternativa, para a compatibilização de tensões das linhas de dados, alguns utilizadores do RFM69CW optam por aplicar divisores de tensão nas linhas de saída do Arduino para o RFM69CW, no entanto a opção por um conversor de níveis garante maior estabilidade e segurança de operação sem danos nos chips.

Como são necessárias as 5 linhas de comunicação acima descritas e o conversor de níveis usado apenas suporta 4, a linha DIO0 de saída do RFM69CW para o microcontrolador é ligada directamente e neste sentido não há à partida problemas de incompatibilidade de tensões pois o microcontrolador percebe perfeitamente os 3,3V como correspondendo ao binário 1 e garantimos que todos os componentes operam dentro das especificações, de forma fiável e sem fumo ...



## Montagem

Neste momento o circuito está montado numa placa de desenvolvimento (*breadboard*), com os fios passados entre os módulos de forma similar à ilustrada abaixo. Digo similar porque os módulos do RFM69CW e do conversor de níveis que usei não estão exactamente na posição ilustrada.

O conversor de níveis, coloquei-lhe os pinos do lado dos componentes o que dá mais jeito na montagem e identificação dos pinos, na ilustração disponível na aplicação Fritzing que usei, o componente só está disponível com os componentes do lado de cima.

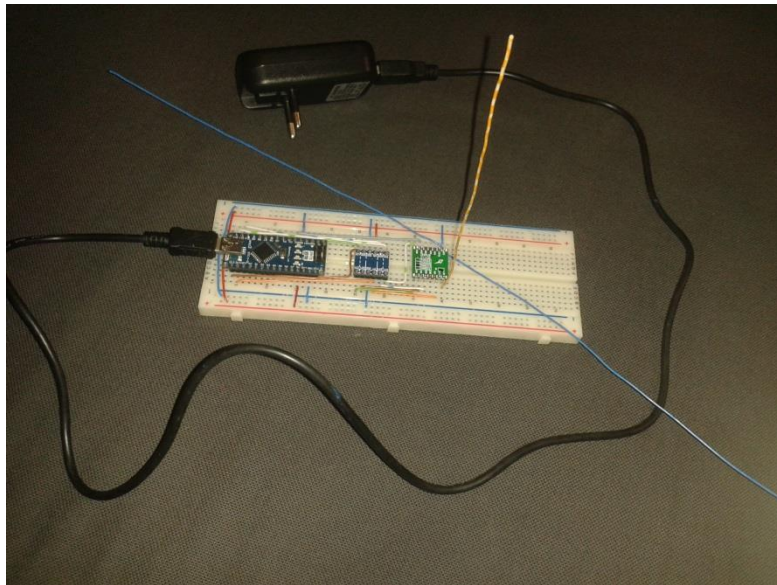
O RFM69CW é um problema diferente, neste caso a ilustração que usei é a do antecessor RFM12B que era a que estava disponível no Fritzing. Infelizmente como qualquer destes módulos tem um espaçamento entre pinos de

2mm que é diferente do clássico 2,54mm (1/10 de polegada), para o compatibilizar com a *breadboard* soldei nos terminais que iria necessitar, pequenos fios que ajetei cuidadosamente com um alicate de pontas para os alinhar com os furos da breadboard.

Obviamente que tirei partido das ligações internas da *breadboard* pelo que alguns fios não seguem todo o percurso ilustrado, apenas parte, porque a restante está dentro da *breadboard*. Tal ocorre por exemplo com as linhas de alimentação de 5V e GND.

Como antena de teste foi usado uma pequena ponta de fio de  $\frac{1}{4}$  de onda, cerca de 17cm para os 433 MHz, com o apoio de duas pontas idênticas na horizontal da placa e ligadas a GND próximo da ligação de antena, de forma a proporcionar um plano de terra mínimo para a configuração típica da antena monopolo. Para ligação a uma antena externa foi usado um pequeno troço de cabo RG58 com dois terminais para a ligação à placa de desenvolvimento e uma ficha BNC para a ligação à baixada da X300N que usei.

Nestas condições tive confirmação de colegas que conseguiram interpretar a comunicação a cerca de 30km!



## Programação

A programação faz-se com a aplicação de desenvolvimento do Arduino que deve ser instalada a partir do *site* oficial.

O programa que usei foi baseado em código de Mark VandeWettering K6HX e nos exemplos da biblioteca de programação do RFM69 desenvolvida por Felix Rusu do LowPowerLab.com, esta biblioteca terá de ser instalada para que a compilação do código seja possível.

No início do código as definições abaixo devem ser substituídas pelos dados considerados adequados de indicativo, localização e frequência de operação.

```
#define HAMCALL "CQ0BEACON"  
#define HAMLOCATOR "IM58MR"  
#define HAMFREQUENCY (432.495+0.00915)
```

Código fonte para programação disponível em: <http://tinyurl.com/ardbalqsp>

## Operação

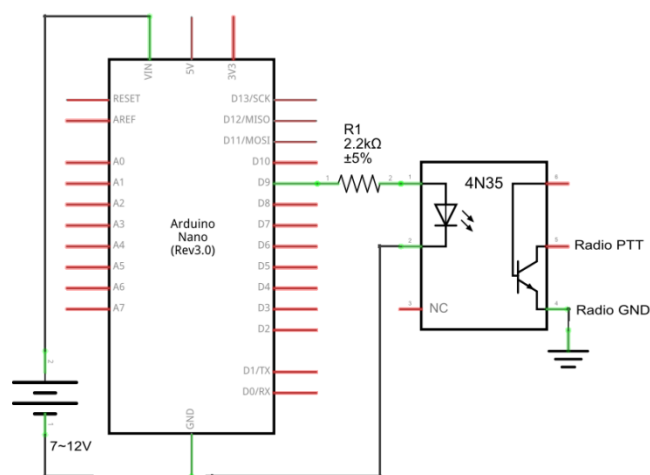
Após o carregamento do código através da ligação USB do PC ao Arduino, a baliza esteve a funcionar com um simples carregador de telemóvel ligado no conector USB.

É possível que o módulo fique uns kHz desviado da frequência esperada, recomendo que após estabilização da temperatura do módulo (observável na porta série/USB e na emissão CW), seja verificado o desvio e aplicado um factor de correcção na programação.

```
///#define NORFM
```

Caso se pretenda usar esta solução como forma de manipulação de uma baliza existente, poderá ser alterada a definição inicial `///#define NORFM`, removendo os `//` no início da linha.

Neste caso, o único módulo necessário é o Arduino a que se recomenda a aplicação do circuito abaixo para segurança na manipulação de PTT de um transceptor existente.



fritzing

## Aspectos legais

Para operação segundo as regras de isenção de licenciamento (não amador), devem ser seguidas as regras indicadas para a banda no anexo 4 do QNAF, salientando-se a necessidade do uso da faixa 433,050 a 434,790 MHz e o limite de potência aparente radiada (PAR) de 10mW, o que exclui uso de qualquer antena exterior.

Para a utilização com as regras de operação de amador, para além da necessidade do utilizador possuir o CAN e operar com as potências e frequências que lhe sejam permitidas, há que salientar o facto da actual legislação não permitir a manutenção de estações de baliza activas por amadores individuais, apenas por associações pelo que, ou está sempre presente quando a baliza estiver activa ou terá de solicitar o apadrinhamento da baliza a uma associação.

## Referências:

Wikipedia SPI: <http://tinyurl.com/wikispi>

Modulo RFM69: <http://tinyurl.com/hoperf69>

Arduíno Nano: <http://tinyurl.com/ardnano>

Conversor de níveis lógicos: <http://tinyurl.com/levelconv>

Biblioteca de programação RFM69 para Arduíno: <http://tinyurl.com/rfm69lib>

Desenho de circuitos Arduíno: <http://fritzing.org/>

ANACOM QNAF: <http://tinyurl.com/qnafdocs>

Código fonte da Baliza QSP/CT7ABP: <http://tinyurl.com/ardbalqsp>