



**UNIVERSIDADE FEDERAL DO RIO DE JANEIRO**  
Instituto de Física  
Programa de Pós-Graduação em Ensino de Física  
Mestrado Profissional em Ensino de Física  
Mestrado Nacional Profissional em Ensino de Física



**MNPEF**

## MONTAGEM DE UMA RADIOSSONDA DE BAIXO CUSTO

Fabiano Pereira de Oliveira  
Helio Salim de Amorim  
Claudine Pereira Dereczynski

Material instrucional associado à  
dissertação de mestrado apresentada ao  
Programa de Pós-Graduação em Ensino de  
Física, Instituto de Física, da Universidade  
Federal do Rio de Janeiro.

Rio de Janeiro  
Novembro de 2016

Caro colega professor, neste trabalho apresenta-se a descrição de como se monta uma radiossonda de baixo custo para utilização com estudantes do ensino médio. Esta atividade foi realizada com alunos do ensino médio em uma escola pública da cidade de Três Rios e revelou-se muito interessante para o grupo. O compartilhamento desta experiência pode ajudar na elaboração de propostas como esta por outros professores.

Este material foi elaborado pensando em fornecer uma montagem fácil para o professor e seus alunos realizarem totalmente dentro de uma sala de aula. Na construção da radiossonda podem ser explorados conhecimentos técnicos, como métodos de soldagem e a teoria envolvendo o funcionamento da radiossonda, como o eletromagnetismo e a ondulatória.

A utilização desta radiossonda e os dados gerados durante um lançamento podem ser aplicados no estudo e investigação de fenômenos atmosféricos de uma forma multidisciplinar e como complemento de aulas sobre temperatura, pressão, estudo dos gases e calor.

## Sumário

O que são radiossondas?.....	4
Motivação .....	5
Materiais necessários.....	5
Montagem .....	7
Programação.....	13
Recepção dos dados.....	22
Lançamento do da radiossonda .....	27
Referências Bibliográficas .....	29

## O que são radiossondas?

Radiossondas são aparelhos meteorológicos utilizados para medir a temperatura, pressão, umidade, direção e velocidade do vento em altitude, que são levados a partir de um balão preenchido com gás hidrogênio ou hélio (Figura 1). Normalmente os balões meteorológicos viajam em condições extremas até a estratosfera, podendo até atingir a mesosfera, os balões lançados pelas estações meteorológicas de altitude chegam aos 25 km de altitude. Durante o voo a radiossonda envia os dados via rádio para a estação em terra.



**Figura 1:** Radiossonda RS92-SGP utilizada na EMA-GL (Estação Meteorológica de Altitude do Galeão).

Os dados meteorológicos coletados através das radiossondas são utilizados nos centros meteorológicos para fornecer um diagnóstico das condições sobre as várias camadas atmosféricas, possibilitando o cálculo de diversos índices de instabilidade, sendo muito úteis na previsão do tempo. Além disso, os dados obtidos nas sondagens alimentam os modelos de previsão numérica do tempo (PNT), compondo, com os demais tipos de dados meteorológicos, as condições iniciais necessárias para integrar tais modelos. No Brasil as radiossondagens são feitas em alguns aeroportos e apenas em dois horários: 0 UTC (21 h local em Brasília) e 12 UTC (9 h local em Brasília).

## Motivação



As radiossondas são equipamentos muito interessantes e o seu uso para o conhecimento do comportamento da atmosfera é muito utilizado na meteorologia como uma ferramenta na previsão das condições do tempo. A utilização dos dados de uma radiossondagem oferece uma riqueza na investigação das variáveis termodinâmicas e, o seu uso em uma aula de Física pode ser muito bem aproveitado.

O maior entrave para a utilização de uma radiossonda na escola básica é o seu preço elevado e, os equipamentos necessários para a recepção dos dados. Pensando nesse problema, esse material propõe uma radiossonda de baixo custo, que pode ser construída por qualquer pessoa interessada, sem demandar conhecimentos avançados em eletrônica.



## Materiais necessários

Para a construção da radiossonda será necessário um micro controlador para projetos embarcados. Em se tratando de uma radiossonda que será lançada de um balão, o peso é muito importante, com isso, a escolha do Arduino Nano V3.0, para a construção.

Para medir a temperatura e pressão, será utilizado o sensor BMP 180, devido ao seu peso reduzido e dupla função. Os dados serão transmitidos via radiofrequência utilizando o módulo RF433MHz. A lista completa com os materiais e ferramentas necessários estão representados na Tabela 1.

Materiais e ferramentas	Descrição
	Arduino Nano V3.0. Valor entre R\$15,00 e R\$25,00.
	BMP180. Valor entre R\$9,00 e R\$20,00.

	<p>Módulo de radiofrequência RF433MHz. Valor entre R\$6,00 e R\$11,00.</p>
	<p>Fio para conexões. Pode-se usar um cabo de rede que tem a espessura ideal para os pinos no Arduino.</p>
	<p>Pote pequeno de isopor. Valor entre R\$2,00 e R\$5,00.</p>
	<p>Conector de bateria 9V. Valor entre R\$1,00 e R\$5,00.</p>
	<p>Bateria 9V. Valor entre R\$4,00 e R\$19,00.</p>
	<p>Tampa de garrafa PET.</p>
	<p>Ferro de solda. Valor entre R\$15,00 e R\$35,00.</p>
	<p>Fio de estanho para soldagem. Valor entre R\$4,00 e R\$9,00.</p>
	<p>Alicate. Valor entre R\$17,00 e R\$60,00.</p>
	<p>Estilete. Valor entre R\$7,00 e R\$20,00.</p>

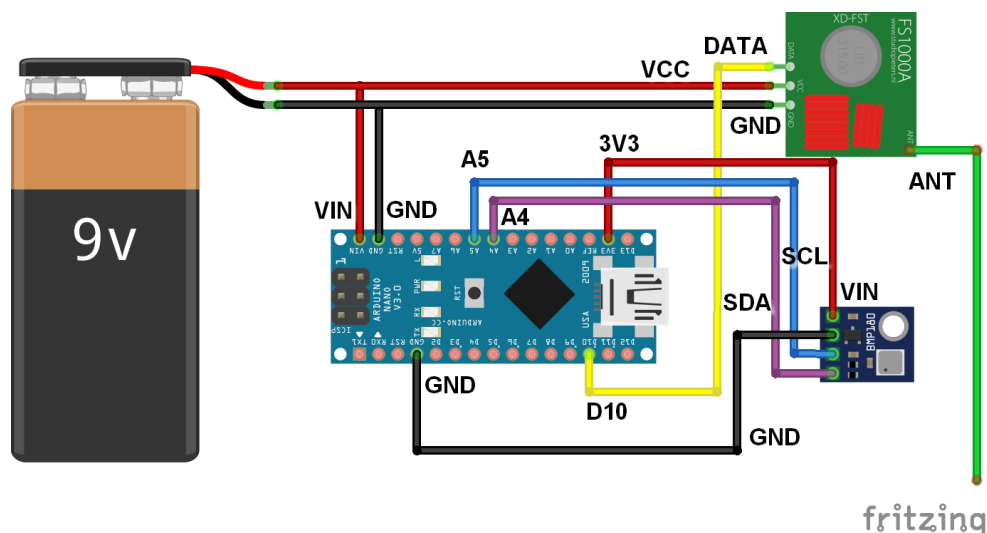
	<p>Cola de isopor. Valor entre R\$2,00 e R\$25,00.</p>
	<p>Fita adesiva. Valor entre R\$2,00 e R\$8,00.</p>

**Tabela 1:** Materiais e ferramentas para a radiossonda. Os valores estimados foram pesquisados no mercado online e lojas fixas na cidade de Três Rios/RJ em outubro de 2016.

Comparando preços e pesquisando lojas é possível construir uma radiossonda por apenas R\$37,00, supondo que ainda seja necessário comprar todas as ferramentas com o melhor preço a radiossonda chega a custar R\$84,00.

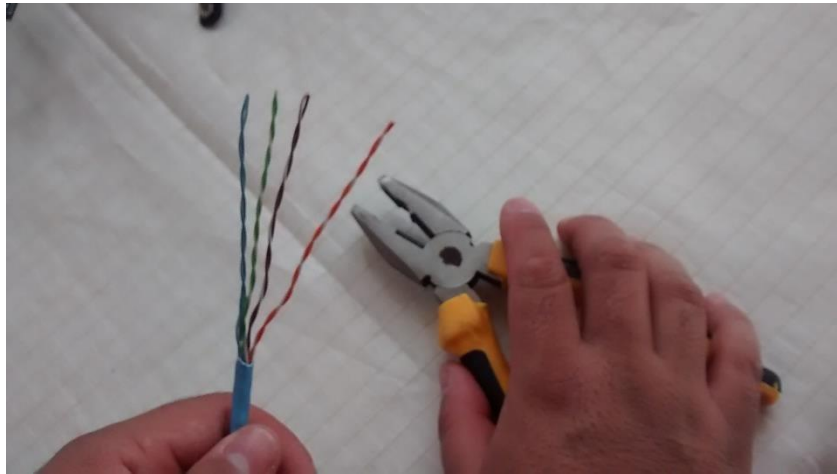
## Montagem

O esquema de montagem dos componentes está representado na Figura 2. Deve-se tomar muito cuidado com as conexões nos terminais para não ligar errado e danificar um componente. A tensão no sensor BMP180 não deve ultrapassar 3,3V e o módulo transmissor, RF 433MHz, deve-se ligar diretamente na bateria de 9V para aproveitar o máximo de potência na transmissão.



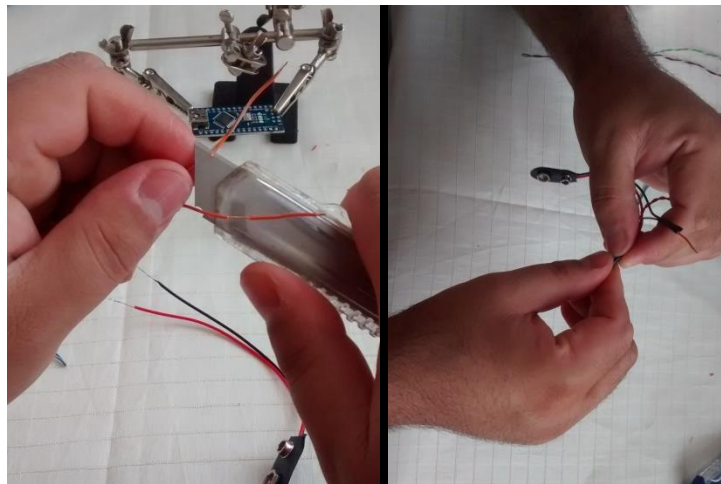
**Figura 2:** Esquema de montagem dos componentes da radiossonda.

Para conectar os componentes no micro controlador, foram utilizados os fios que compõe um cabo de rede de computadores, como mostra a Figura 3.



**Figura 3:** Fios para conexões obtidos de um cabo de rede

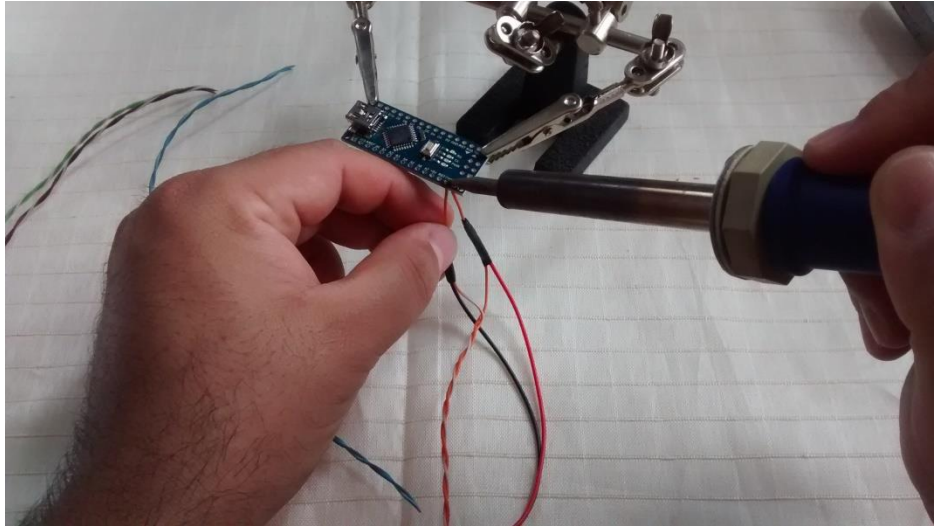
Como o módulo de radio frequência é conectado direto na bateria, deve-se fazer uma ligação de dois fios no *plug* da bateria (Figura 4). Lembre-se de isolar para evitar curtos.



**Figura 4:** Conexão dos fios do RF433 MHz no *plug* da bateria.

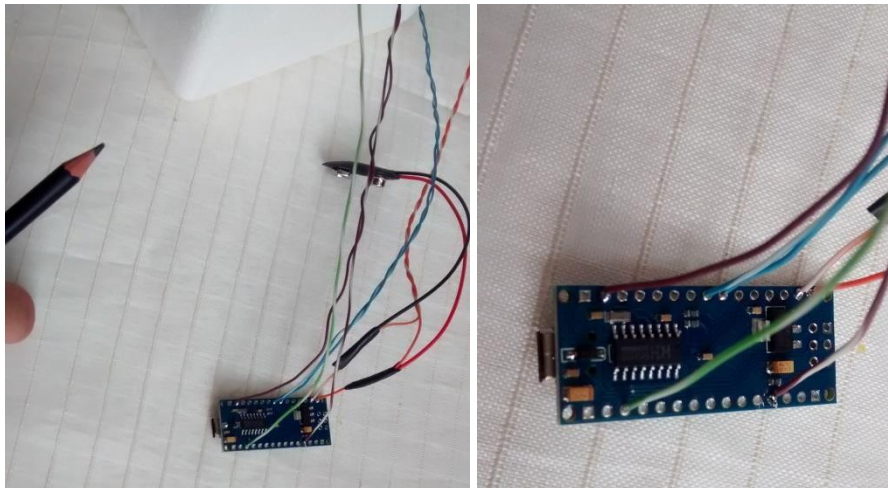
Para soldar os fios na placa do Arduino Nano, é aconselhável ter um suporte para facilitar o trabalho, lembre-se que deve utilizar somente a quantidade de solda suficiente para prender o fio, solda em excesso pode provocar curto na placa. Dê preferência a um ferro de solda com ponta fina, como mostra Figura 5.





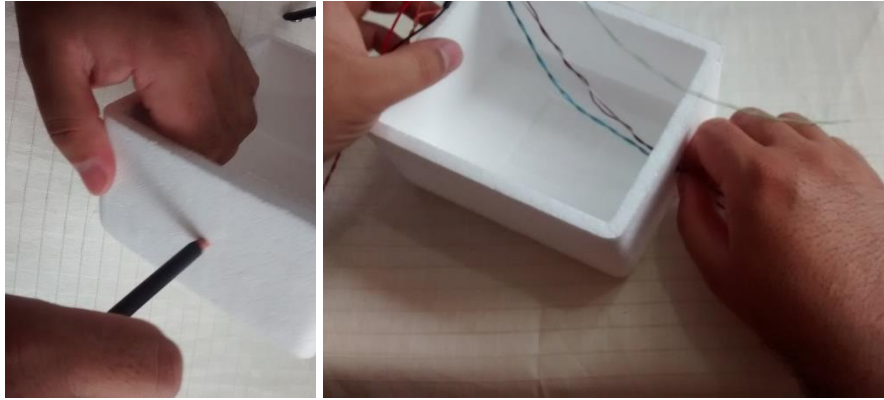
**Figura 5:** Soldando os fios na placa Arduino Nano.

Depois de soldar todos os fios, como mostrado no diagrama, a aparência final da placa deve ser parecida com a Figura 6.



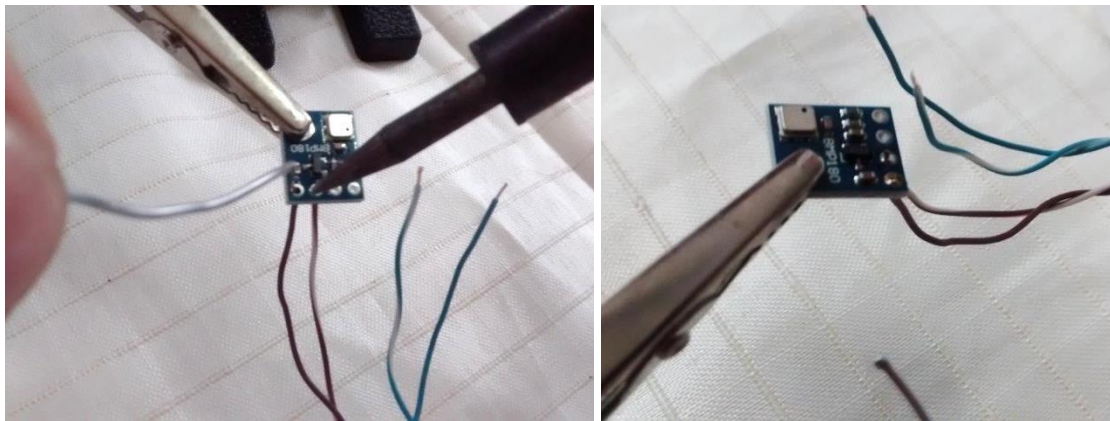
**Figura 6:** Arduino Nano com todos os cabos conectados.

Antes de soldar o sensor BMP180, deve-se fazer um furo no pote de isopor, para passar os quatro fios referentes ao sensor. Para o furo pode-se utilizar um lápis, como mostra a Figura 7.



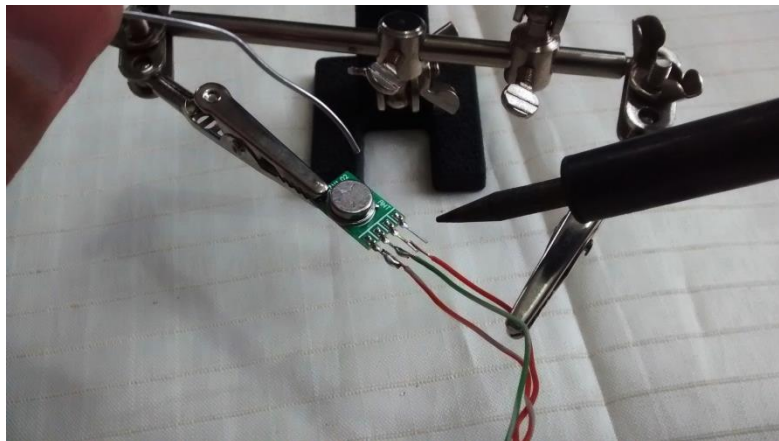
**Figura 7:** Furo no pote de isopor para passar os fios do sensor BMP180.

Com o fio posicionado e, já passando pelo orifício, pode-se soldar o BMP180 (Figura 8). Deve-se ter muito cuidado nesse processo para não danificar o sensor.



**Figura 8:** Soldando o sensor BMP180.

Em seguida, precisamos soldar o módulo de radiofrequência, RF 433 MHz, verifique os pinos corretos antes de soldar, existem algumas variações desse módulo onde os pinos são diferentes (Figura 9).



**Figura 9:** Soldando o módulo RF 433 MHz.

Como o sensor BMP180 deve ficar no exterior do pote, em contato direto com a atmosfera, deve-se montar um abrigo de proteção contra a radiação solar. Para esse objetivo foi utilizado uma tampinha de refrigerante, como mostra a Figura 10.



**Figura 10:** Construindo um abrigo contra radiação solar para o sensor BMP180.

Para fixar o abrigo do sensor no pote, deve-se marcar com a própria tampinha a sua localização, forçando até deixar uma marca, com profundidade de 1 mm a 2 mm, como mostra a Figura 11.



**Figura 11:** Marcação da posição do abrigo do sensor BMP180.

Depois de marcado a posição, pode-se utilizar uma cola de isopor para fixar bem a tampinha no local marcado.

Para a construção da antena, pode-se utilizar um fio rígido fino, como encontrado dentro do cabo de rede. O tamanho do fio para funcionar como antena na

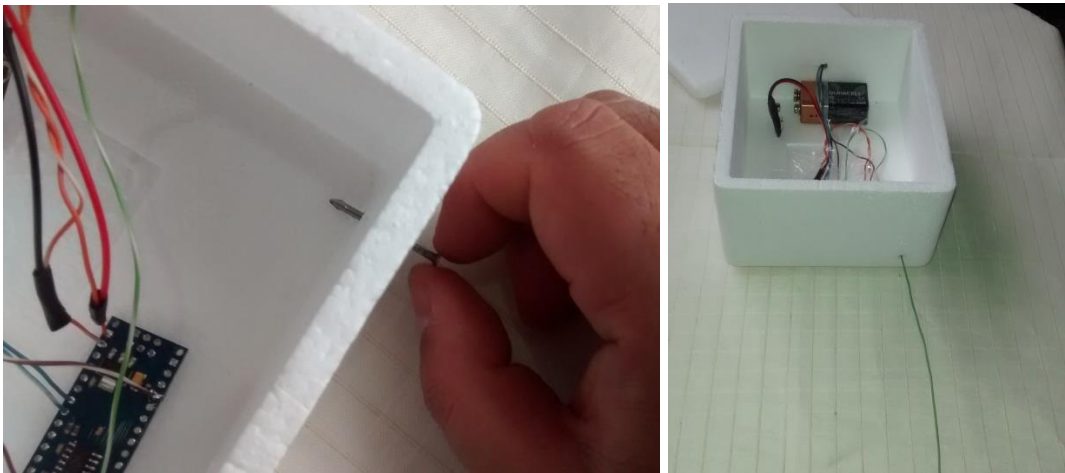
radiossonda deve ser calculado usando conhecimentos básicos de Física, onde ( $c$ ) é a velocidade de propagação, ( $f$ ) a frequência da onda e ( $\lambda$ ) o comprimento de onda.

$$\lambda = \frac{c}{f}$$

$$\lambda = \frac{3.10^8 m/s}{433.10^6 MHz} \cong 0,6928 m$$

Não é necessário utilizar um fio com 69 cm de comprimento. A antena utilizada neste trabalho é uma monopolo de  $\frac{1}{4}\lambda$ , assim, o seu comprimento deve ter 17,3 cm.

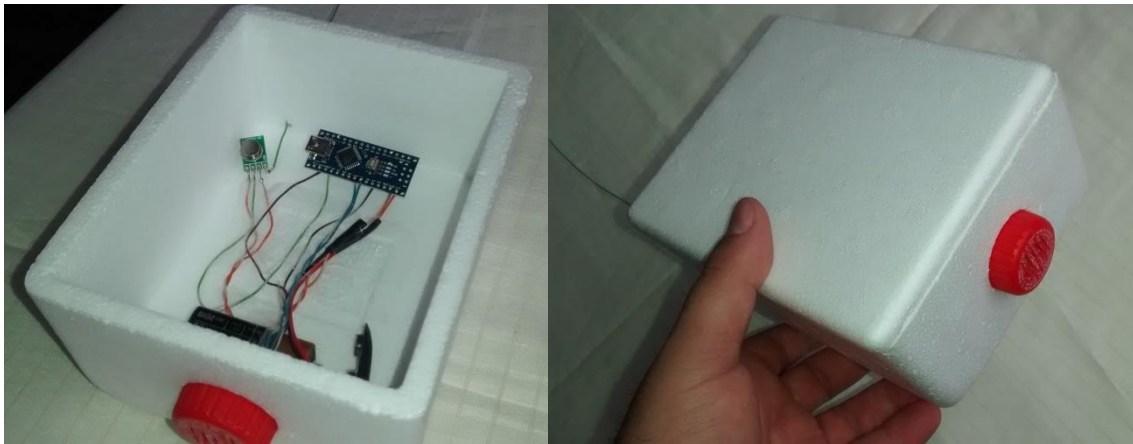
Depois de cortado o fio e, soldado no módulo RF 433 MHz, é necessário fazer um furo no pote para a sua passagem. O furo pode ser feito com um prego, como mostrado na Figura 12.



**Figura 12:** Furo no pote para a passagem da antena.

A partir desse momento, é só acomodar os componentes dentro do pote e, prender com fita isolante para evitar movimentos indesejados, principalmente a bateria, devido a possíveis turbulências. O uso de cola ou silicone não é aconselhável devido ao acréscimo de peso à radiossonda. A Figura 13 mostra o aspecto final da radiossonda de baixo custo.





**Figura 13:** Radiossonda pronta, com todos os seus componentes.

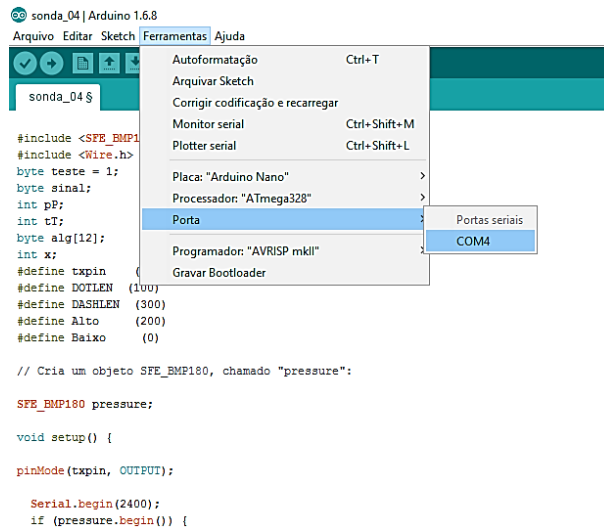
## Programação

Com o equipamento montado, é necessário programar o microcontrolador para a leitura dos dados do sensor e posterior envio via radiofrequência para a estação em terra onde será gravado. É possível acrescentar um *shield*<sup>1</sup> para gravar os dados dentro da radiossonda, mas nesse projeto não está previsto, devido à necessidade de busca da radiossonda depois de lançada, o que acarreta também um acréscimo de um módulo GPS para o rastreamento.

A programação do Arduino Nano é realizada utilizando a IDE do Arduino, que deve estar configurada para se comunicar via porta serial. A bateria da radiossonda não deve estar ligada durante o processo de programação. Ao conectar o cabo USB, deve-se clicar em “Ferramentas” e selecionar a placa Arduino Nano e, conferir se foi adicionado uma porta serial para a comunicação da placa (Figura 14).

---

<sup>1</sup> "Shields" é o nome dado às placas de expansão de *hardware* que encaixam na placa Arduino principal. Através dos *shields*, uma placa Arduino pode ser melhorada ainda mais! Os circuitos contidos nos diversos *shields* contêm uma eletrônica que adiciona funções que a placa principal não possui. [www.robotizando.com.br/shields\\_index.php](http://www.robotizando.com.br/shields_index.php). Acesso em 14/10/2016.



**Figura 14:** Configuração da porta serial na IDE do Arduino.

Para que o Arduino Nano interprete os dados do sensor de temperatura e pressão, deve-se instalar a sua biblioteca<sup>2</sup> dentro da IDE.

Com a biblioteca instalada e a placa conectada corretamente na IDE deve-se inserir o código representado a seguir.

#### Início do código.

```
#include <SFE_BMP180.h>
#include <Wire.h>
byte teste = 1;
byte sinal;
int pP;
int tT;
byte alg[12];
int x;
#define txpin (10)
#define DOTLEN (100)
#define DASHLEN (300)
#define Alto (200)
#define Baixo (0)

// Cria um objeto SFE_BMP180, chamado "pressure":
SFE_BMP180 pressure;

void setup() {
  pinMode(txpin, OUTPUT);

  Serial.begin(2400);
  if (pressure.begin()) {
    Serial.println("BMP180 inicializado com sucesso");}
  //else
```

<sup>2</sup> Biblioteca do sensor BMP 180 disponível em <https://learn.sparkfun.com/tutorials/bmp180-barometric-pressure-sensor-hookup-/installing-the-arduino-library>. Acesso em 14/10/2016.

```

    //{
    //    Serial.println("BMP180 falhou inicializar\n\n");
    //while(1); // Pause forever.
    //}

}

void loop()
{
    double T,P;    // lê novo valor de pressão e temperatura, ...

    P = getPressure();
    T = getTemperature();

    //...separa em seus algoritmos e armazena no array alg[].

    for(int x=0; x<12; x++){alg[x]=0;}

    if (T<0){sinal = 2;}
    else {sinal=1;}

    pP = int(P);
    tT = int(T*100);

    int i=11;

    do {alg[i]=pP%10;
    i=i-1;
    pP=int(pP/10);}
    while (pP>0);

    i=5;
    do {alg[i]=tT%10;
    i=i-1;
    tT = int(tT/10);}
    while (tT>0);

    alg[0] = sinal; //variável que transporta o sinal da temperatura.

    //Transmite os dados:

    for (int x = 0; x<12; x++) {

        switch (alg[x]) {        //dah-dah-dah-dah-dah
        case 0:
            analogWrite(txpin, Alto);
            delay(DASHLEN);
            analogWrite(txpin, Baixo);
            delay(DOTLEN);

            analogWrite(txpin, Alto);
            delay(DASHLEN);
            analogWrite(txpin, Baixo);
            delay(DOTLEN);

            analogWrite(txpin, Alto);
            delay(DASHLEN);
            analogWrite(txpin, Baixo);
            delay(DOTLEN);

```

```

analogWrite(txpin, Alto);
delay(DASHLEN);
analogWrite(txpin, Baixo);
delay(DOTLEN);

analogWrite(txpin, Alto);
delay(DASHLEN);
analogWrite(txpin, Baixo);
delay(DASHLEN);
break;

case 1: //dit-dah-dah-dah-dah

analogWrite(txpin, Alto);
delay(DOTLEN);
analogWrite(txpin, Baixo);
delay(DOTLEN);

analogWrite(txpin, Alto);
delay(DASHLEN);
analogWrite(txpin, Baixo);
delay(DOTLEN);

analogWrite(txpin, Alto);
delay(DASHLEN);
analogWrite(txpin, Baixo);
delay(DOTLEN);

analogWrite(txpin, Alto);
delay(DASHLEN);
analogWrite(txpin, Baixo);
delay(DOTLEN);

analogWrite(txpin, Alto);
delay(DASHLEN);
analogWrite(txpin, Baixo);
delay(DASHLEN);
break;

case 2: //dit-dit-dah-dah-dah

analogWrite(txpin, Alto);
delay(DOTLEN);
analogWrite(txpin, Baixo);
delay(DOTLEN);

analogWrite(txpin, Alto);
delay(DOTLEN);
analogWrite(txpin, Baixo);
delay(DOTLEN);

analogWrite(txpin, Alto);
delay(DASHLEN);
analogWrite(txpin, Baixo);
delay(DOTLEN);

analogWrite(txpin, Alto);
delay(DASHLEN);
analogWrite(txpin, Baixo);
delay(DOTLEN);

```



```

    analogWrite(txpin, Alto);
    delay(DASHLEN);
    analogWrite(txpin, Baixo);
    delay(DASHLEN);
    break;

case 3:    //dit-dit-dit-dah-dah

    analogWrite(txpin, Alto);
    delay(DOTLEN);
    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DOTLEN);
    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DOTLEN);
    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DASHLEN);
    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DASHLEN);
    analogWrite(txpin, Baixo);
    delay(DASHLEN);
    break;

case 4: //dit-dit-dit-dit-dah

    analogWrite(txpin, Alto);
    delay(DOTLEN);
    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DOTLEN);
    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DOTLEN);
    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DOTLEN);
    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DASHLEN);
    analogWrite(txpin, Baixo);
    delay(DASHLEN);
    break;

```

```
case 5:    //dit-dit-dit-dit-dit
```

```
    analogWrite(txpin, Alto);
    delay(DOTLEN);
    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DOTLEN);
    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DOTLEN);
    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DOTLEN);
    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DOTLEN);
    analogWrite(txpin, Baixo);
    delay(DASHLEN);
    break;
```

```
case 6:    //dah-dit-dit-dit-dit
```

```
    analogWrite(txpin, Alto);
    delay(DASHLEN);
    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DOTLEN);
    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DOTLEN);
    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DOTLEN);
    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DOTLEN);
    analogWrite(txpin, Baixo);
    delay(DASHLEN);
    break;
```

```
case 7:    //dah-dah-dit-dit-dit
```

```
    analogWrite(txpin, Alto);
    delay(DASHLEN);
```

```

    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DASHLEN);
    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DOTLEN);
    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DOTLEN);
    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DOTLEN);
    analogWrite(txpin, Baixo);
    delay(DASHLEN);
    break;

case 8:    //dah-dah-dah-dit-dit

    analogWrite(txpin, Alto);
    delay(DASHLEN);
    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DASHLEN);
    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DASHLEN);
    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DOTLEN);
    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DOTLEN);
    analogWrite(txpin, Baixo);
    delay(DASHLEN);
    break;

case 9:    //dah-dah-dah-dah-dit

    analogWrite(txpin, Alto);
    delay(DASHLEN);
    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DASHLEN);

```

```

    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DASHLEN);
    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DASHLEN);
    analogWrite(txpin, Baixo);
    delay(DOTLEN);

    analogWrite(txpin, Alto);
    delay(DOTLEN);
    analogWrite(txpin, Baixo);
    delay(DASHLEN);
    break;
}
Serial.print(alg[x]);
}

Serial.println();

delay(10000);
}

//-----
//Função que lê a pressão.

double getPressure()
{
    char status;
    double T,P;

    // A função status retorna 0 se tiver falha e 1 se tiver
    sucesso.

    status = pressure.startTemperature();
    if (status != 0)
    {
        // Espera a medida se completar:
        delay(status);

    status = pressure.getTemperature(T);
    if (status != 0)
        {

        status = pressure.startPressure(3);
        if (status != 0)
            {
            // Espera a medida se completar:
            delay(status);

        status = pressure.getPressure(P,T);
        if (status != 0)
            {
            return(P);
            }
        }
    }
}

```

```

        else Serial.println("Erro tentando medir a pressão\n");
    }
    else Serial.println("Erro ao iniciar a medida da pressão\n");
}
    else Serial.println("Erro tentando medir a temperatura\n");
}
    else Serial.println("Erro ao iniciar a medida de
temperatura\n");
}

//-----
//Função que lê a temperatura.

double getTemperature()
{
    char status;
    double T,P;

    // A função status retorna 0 se tiver falha e 1 se tiver
sucesso.

    status = pressure.startTemperature();
    if (status != 0)
    {
        // Espera a medida se completar:
        delay(status);

        status = pressure.getTemperature(T);

        if (status != 0)
        {

            status = pressure.startPressure(3);
            if (status != 0)
            {
                // Espera a medida se completar:
                delay(status);

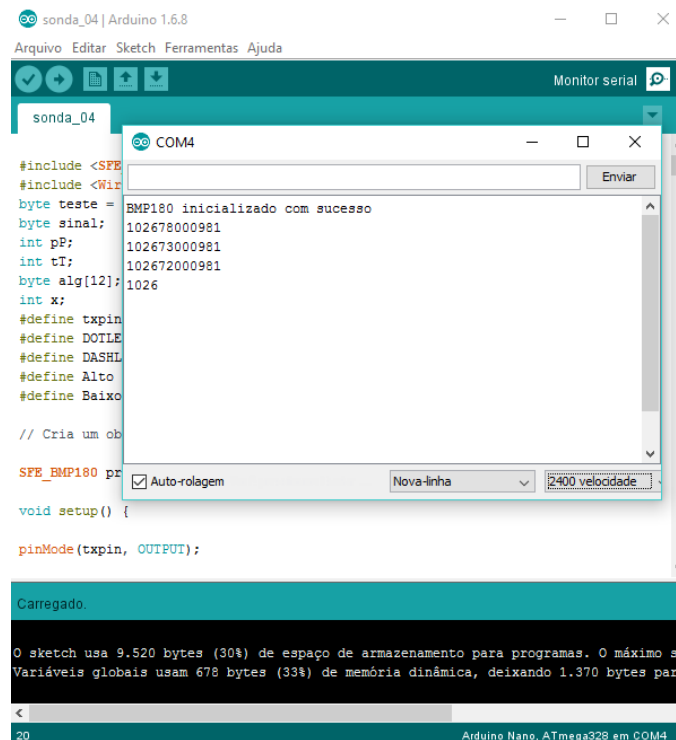
                status = pressure.getPressure(P,T);
                if (status != 0)
                {
                    return(T);
                }
            }
            else Serial.println("Erro tentando medir a pressão\n");
        }
        else Serial.println("Erro ao iniciar a medida da pressão\n");
    }
    else Serial.println("Erro tentando medir a temperatura\n");
}
    else Serial.println("Erro ao iniciar a medida de
temperatura\n");
}
}

```

Fim do Código.

Depois de inserir o código na IDE, deve-se clicar em “upload” para enviar a programação à placa. Por fim, completado o processo com sucesso, é possível verificar

se a radiossonda está funcionando corretamente clicando em “Monitor Serial” e, se estiver funcionando normalmente a tela deve ser parecida com a Figura 15.



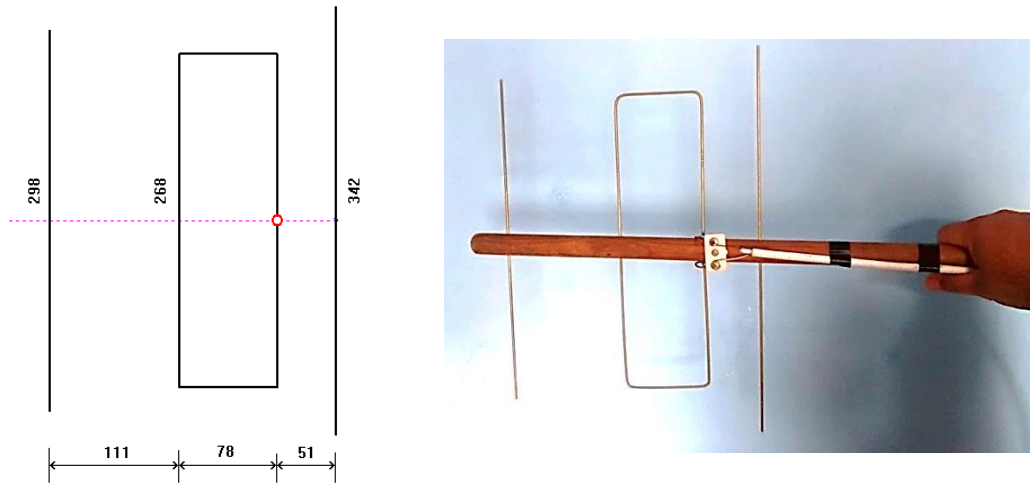
**Figura 15:** Monitor Serial da IDE do Arduino verificando o funcionamento da radiossonda.

Em se tratando de comunicação a longa distância é comum se deparar com o problema do alcance na transmissão, visto que não é possível utilizar equipamento com uma potência alta em função da proibição da legislação vigente. O módulo de rádio frequência utilizado neste trabalho é projetado para comunicação via *wireless* com um alcance máximo de 100 m, como recomenda o fabricante, mas desconsiderando o protocolo de comunicação do módulo e utilizando uma das formas mais antigas de comunicação, código Morse, é possível aumentar o alcance surpreendentemente. Dessa forma o código acima lê os dados do sensor de temperatura e pressão, separa em algarismos e depois envia, via código Morse, na frequência de 433 MHz.

## Recepção dos dados

A recepção dos dados começa com uma antena apropriada, visto que a potência de transmissão da radiossonda é baixa e a distância é longa. Pode-se ter um ganho razoável com uma antena direcional do tipo Yagi LFA de três elementos, construída

com varetas de solda amarela de 3,2 mm de diâmetro. Medidas para a construção da antena pode ser encontrada na Figura 16.



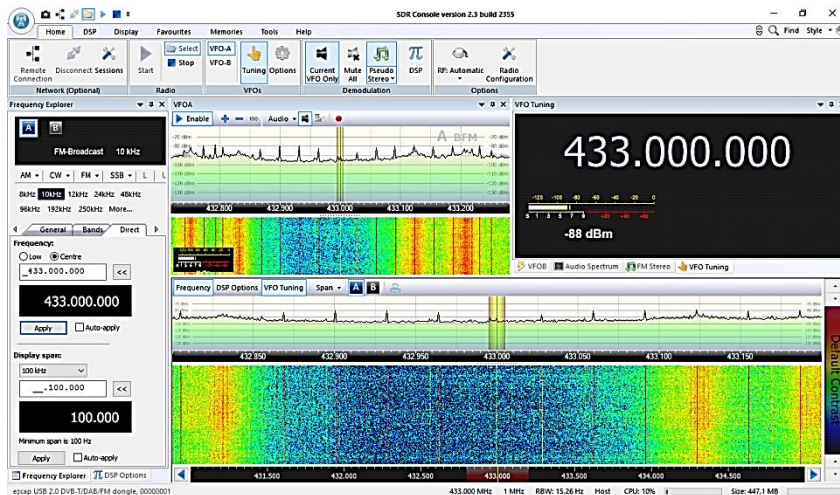
**Figura 16:** Configuração da antena Yagi LFA para captar a frequência 433 MHz. Medidas em mm. Fonte: <http://www.qsl.net/py4zbx/antenas/LFA.htm>. Acesso em 22/07/2016.

A antena deve ser conectada em um rádio que seja capaz de receber a frequência de 433 MHz. Uma ótima opção é o *dongle* RTL-SDR (Figura 17), devido a sua vasta faixa de frequência para recepção. Para ter acesso as configurações do rádio é preciso utilizar um *software* de controle chamado SDR-Rádio<sup>3</sup> (Figura 18).



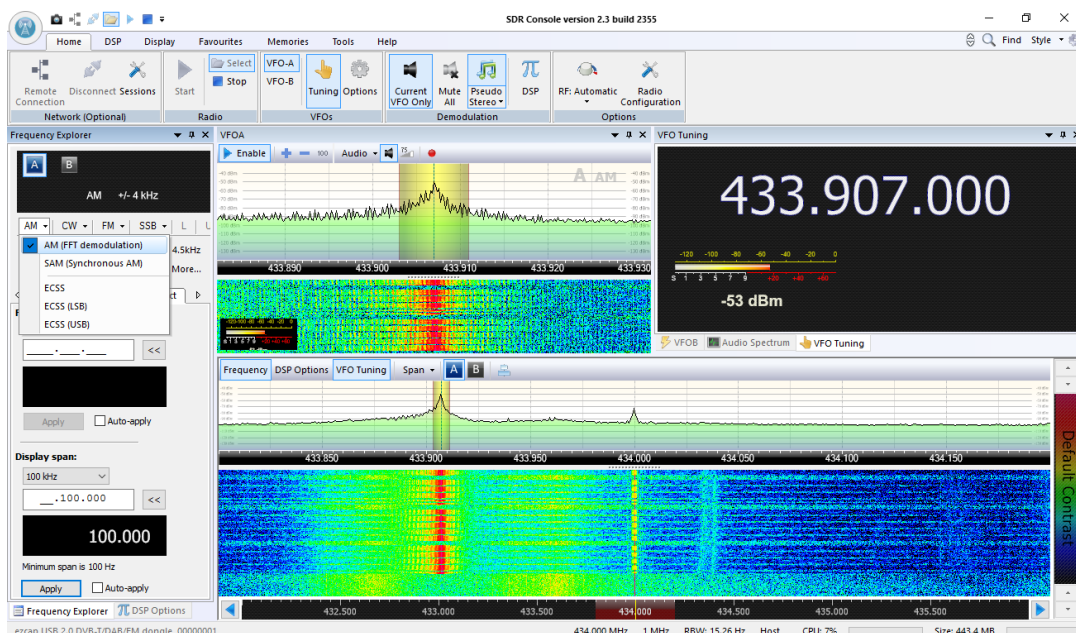
**Figura 17:** Dongle RTL – SDR

<sup>3</sup> SDR-Radio <http://sdr-radio.com/>. Acesso em 23/07/2016.



**Figura 18:** SDR-Rádio sintonizado na frequência 433 MHz

Pode-se verificar que há algumas interferências na proximidade dessa frequência, isso é normal, pois ela é usada na maioria dos equipamentos domésticos e alarme de carro. Ao ligar a radiossonda na bateria, a transmissão se inicia e, já será possível observar o espectro no *software* do radio. Para melhorar a qualidade da recepção, mude no canto esquerdo do SDR-Rádio a configuração para “AM (FFT demodulation)” e, procure a melhor sintonia no espectro de frequência (Figura 19).



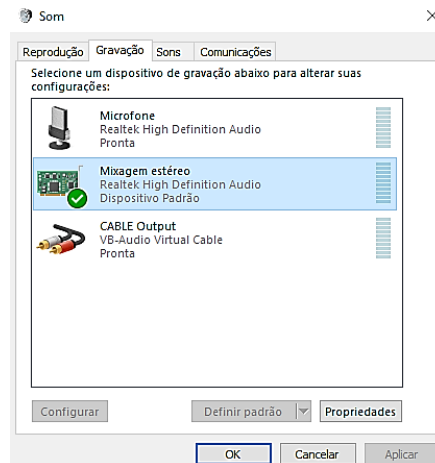
**Figura 19:** SDR-Rádio configurado para receber o sinal mais limpo da radiossonda.

Com o áudio da telemetria sendo recebido, é necessário um programa para decodificar o código Morse e mostrar as leituras de pressão e temperatura. Um



programa gratuito e muito eficiente é o EhoCW<sup>4</sup>, a sua versão *freeware* oferece muitos recursos e para o objetivo deste trabalho atende muito bem ao propósito.

Antes de entrar no EhoCW, deve-se configurar o dispositivo de gravação para “Mixagem estéreo” para o seu funcionamento correto (Figura 20).



**Figura 20:** Configuração do áudio para utilizar o EhoCW.

Dentro do EhoCW, clicar na aba “Decoder” e, deixar todas as opções marcadas na parte inferior, principalmente “Decoder”, como mostra a Figura 21. Com o áudio sendo executado pelo SDR-Rádio, o EhoCW começará a interpretar o código Morse e, mostrar os valores em sua tela.

---

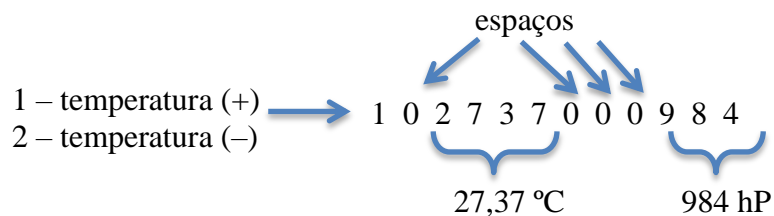
<sup>4</sup> EhoCW é um *software* muito usado por radioamadores em treinamento de código Morse. Disponível em <http://www.f8eho.net/>. Acesso em 20/07/2016.



**Figura 21:** EhoCW decodificando os dados da radiossonda

É possível observar que alguns dados foram perdidos devido a má interpretação pelo EhoCW, esse problema normalmente é causado por alguma interferência na recepção, por isso a importância em ficar em um local mais isolado, sem influência de campos eletromagnéticos. Mesmo com interferência é possível aproveitar a maior parte dos dados.

Cada transmissão da radiossonda é feita em pacotes com doze algarismos, separados por um espaço; o primeiro representa o sinal da temperatura (1 para temperatura positiva e 2 para negativa), seguido pela temperatura, sem a vírgula de separação das casas decimais e, por fim a pressão, como mostra a Figura 22.



**Figura 22:** Pacote de dados enviado pela radiossonda e, o padrão utilizado.

Após a decodificação pelo EhoCW, é possível gravar os dados em arquivo texto e, posterior análise em programas gráficos para o estudo.

## Lançamento da radiossonda

O Comando da Aeronáutica disciplina o uso na Portaria CIRCEA 63-3 “Procedimentos operacionais referentes ao lançamento de balões meteorológicos”.

### COMPETE AOS ÓRGÃOS DE METEOROLOGIA

As Estações de radiossonda operadas pelo Comando da Aeronáutica deverão informar ao órgão de tráfego aéreo local:

- a) o horário de lançamento, com antecedência mínima de 30 minutos;
- b) a cor do balão meteorológico;
- c) outras informações julgadas úteis à segurança da navegação aérea tais como a posição do balão meteorológico durante a subida, quando possível;
- d) o horário do momento em que o balão meteorológico se romper;
- e) caso haja um segundo lançamento, o novo horário previsto, com a devida antecedência; e
- f) quando não for realizado o lançamento, por qualquer motivo.

NOTA: Em casos eventuais de radiossondagens realizadas em outros horários, os procedimentos devem ser repetidos como se fossem as do horário padrão.  
(BRASIL, 2013)

Com as devidas precauções e, com a autorização, se necessário, deve-se escolher um lugar aberto, sem linhas de transmissão e preferencialmente fora do trajeto de linhas aéreas<sup>5</sup>. A radiossonda deve ser lacrada com fita isolante e identificada com um adesivo contendo os dados do projeto, autor, horário, dia de lançamento e um telefone, caso seja encontrada para possível devolução.

O balão meteorológico é vendido no comércio internacional, mas o propósito deste trabalho é uma radiossonda de baixo custo e, o balão pode ser substituído por balão de festas, conhecido como “Big Ballon” de número 350 (Figura 23).

---

<sup>5</sup> O aplicativo “Flightradar24” para *smartphone*, permite verificar em tempo real a trajetória de voos comerciais e a posição das aeronaves no globo terrestre.



**Figura 23:** Balão meteorológico. À esquerda balão utilizado na Estação Meteorológica de Altitude do Galeão, à direita o balão de festas (nº350), utilizado neste projeto.

Para inflar o balão, utiliza-se o gás Hélio por questões de segurança devido ao seu estado inerte. Algumas lojas alugam o cilindro cheio para festas, outra opção é comprar o cilindro preparado em lojas especializadas. É altamente recomendado utilizar o cilindro com um regulador de pressão para um enchimento seguro e lento (Figura 24).



**Figura 24:** Cilindro com gás Hélio utilizado no projeto.

Aconselha-se a utilização de um paraquedas, conectado à radiossonda e o balão, a fim de evitar acidentes no processo de descida da radiossonda.



**Figura 25:** Paraquedas utilizado para a segurança na descida da radiossonda.

Com a radiossonda amarrada e lacrada, o paraquedas de segurança conectado e o balão cheio, basta conferir se a telemetria da radiossonda no computador está funcionando e, soltar o balão em um local livre de obstáculos.

## Referências Bibliográficas

ALVARENGA, B.; MÁXIMO, A. **Curso de Física**. 6ª. ed. São Paulo: Scipione, v. 3, 2007.

ALVARENGA, B.; MÁXIMO, A. **Curso de Física**. 6ª. ed. São Paulo: Scipione, v. 2, 2007.

ANATEL. Anatel. **Anatel**, 2016. Disponível em:

<<http://www.anatel.gov.br/Portal/exibirPortalNivelDois.do?codItemCanal=1764&nomeVisao=Cidad%E3o&nomeCanal=Comunica%E7%F5es%20Via%20R%E1dio&nomeItemCanal=Ondas%20de%20R%E1dio>>. Acesso em: 09 jul. 2016.

ARDUINO. **Arduino**, 2016. Disponível em: <<https://www.arduino.cc/>>. Acesso em: 07 jun. 2016.

AZEVEDO, M. C. P. S. Ensino por Investigação: Problematizando as Atividades em Sala de Aula. In: CARVALHO, A. M. P. **Ensino de Ciências - unindo a pesquisa e a prática**. São Paulo: Cengage Learning Editores, 2004.

BRASIL, M. D. D. C. D. A. **Proteção ao voo: PROCEDIMENTOS OPERACIONAIS REFERENTES AO LANÇAMENTO DE BALÕES METEOROLÓGICOS**. Ministério da Defesa. [S.l.]. 2013.

CARVALHO, Á. G. D.; BADINHAN, F. D. C. **Eletrônica: Telecomunicações**. São Paulo: Fundação Padre Anchieta, v. 5, 2011.

CARVALHO, A. M. P. D.; AZEVEDO, M. C. P. S. D. **Ensino de Ciências - Unindo a Pesquisa e a Prática**. São Paulo: Cengage Learning Editores, 2004.

MCRBERTS, M. **Arduino Básico**. São Paulo: Novatec, 2011.

MEDEIROS, J. C. D. O. **Princípios de Telecomunicações: Teoria e Prática**. 2ª. ed. São Paulo: Érica, 2007.

NUSENZVEIG, H. M. **Curso de Física Básica**. 1ª. ed. São Paulo: Edgard Blücher, v. 3, 1997.