

Um marégrafo ultrassônico baseado na placa Arduino para investigação do fenômeno das marés^{+,*}

R. G. Soares¹

Professor de Física do Ensino Médio da rede particular de ensino

H. S. Amorim¹

Instituto de Física – Universidade Federal do Rio de Janeiro

Rio de Janeiro – RJ

Resumo

Neste trabalho apresentamos um projeto de construção de um marégrafo baseado num telêmetro ultrassônico controlado por uma placa Arduino Uno. O projeto é de fácil construção e de baixo custo e tem por objetivo viabilizar o estudo do fenômeno das marés oceânicas integrando atividades investigativas com a prática experimental. O fenômeno das marés se mostra uma excelente oportunidade para o estudo da gravitação universal de Isaac Newton e do estudo do movimento orbital do sistema Terra – Lua – Sol, dos quais é um resultado direto. Neste artigo, focalizamos basicamente a montagem experimental e apresentamos alguns resultados preliminares. Reservamos para um próximo artigo sugestões de aplicações em eventos investigativos bem definidos envolvendo o modelo newtoniano para o fenômeno das marés.

Palavras-chave: *Maré; Marégrafo; Telêmetro Ultrassônico; Arduino.*

Abstract

In this work, we present a project to construct a tide gauge based on an ultrasonic rangefinder controlled by an Arduino Uno board. The project is easy to build and of low cost and aims to make feasible the study of the phenomenon of ocean tides, integrating research activities and experimental practice. The tidal phenomenon is an excellent opportunity

⁺ An ultrasonic tide gauge based on the Arduino board for a tidal phenomenon investigation

^{*} *Recebido: maio de 2019.*

Aceito: junho de 2020.

¹ E-mails: rafaelguedessoares@yahoo.com.br; hsalim@if.ufrj.br

for the study of Isaac Newton's universal gravitation and the study of the orbital motion of the Earth – Moon – Sun system, of which it is a direct result. In this article, we focus on the experimental set up and preliminary results. We reserve for a next article suggestions of applications in well-defined investigative events involving the Newtonian model for the tidal phenomenon.

Keywords: *Tides; Tide Gauge; Ultrasonic Rangefinder; Arduino.*

I. Introdução

Neste trabalho apresentamos um projeto de construção de um marégrafo baseado num telêmetro ultrassônico controlado por uma placa Arduino Uno. O projeto tem por objetivo o estudo do fenômeno de marés para turmas do Ensino Médio, em uma sequência de ensino investigativo, privilegiando a atividade experimental e instrumental.

As marés, como se manifesta nos mares e oceanos da Terra, é muito importante para inúmeras atividades humanas, particularmente as atividades de navegação e pesca, e são essenciais para vários processos de manutenção da vida nos mais diversos ambientes marinhos. Os registros históricos indicam que o homem, já há milênios, reconhece essa importância. A regularidade temporal em que sucedem a elevação e o abaixamento do nível do mar e sua relação com o movimento da Lua e do Sol sempre foram motivo de curiosidade e encantamento. Entretanto, somente com o advento da mecânica newtoniana, da teoria da gravitação universal e do cálculo, a partir do século XVII, foi possível um entendimento satisfatório sobre a dinâmica das marés. As contribuições de Isaac Newton e Pierre-Simon Laplace foram decisivas para a visão atual sobre o fenômeno.

Por ser um fenômeno de base gravitacional, as marés oferecem um cenário muito rico para o estudo da teoria da gravitação universal de Newton nos níveis Fundamental e Médio, pois são como 'gravitação em ação', alterando constante e periodicamente o cenário de nossas costas, praias e lagunas.

Para não estender demasiadamente esta apresentação, separamos o conteúdo deste trabalho em duas partes. Nessa primeira, nos dedicamos à descrição da parte instrumental somente. Um segundo trabalho está sendo encaminhado com aplicações dos dados coletados com o marégrafo ultrassônico em atividades investigativas sobre o fenômeno das marés. Alguns artigos relativamente recentes relacionados ao tema das marés podem ser encontrados em Carvalho (2014) e Silveira (2003).

No que se segue, vamos iniciar descrevendo esquematicamente o projeto do marégrafo para, na sequência, apresentarmos detalhes práticos de construção e instalação. Concluímos com a apresentação de alguns resultados preliminares.

II. Um marégrafo ultrassônico baseado na placa Arduino

O marégrafo que vamos descrever a seguir foi projetado para ser aplicado basicamente em atividades educacionais. Um dos problemas principais de sua aplicação em escolas é a dificuldade natural de se conseguir um lugar para a observação das marés. Dificilmente uma escola dispõe de um acesso direto ao mar em suas dependências e, para a aplicação do projeto, será necessário estabelecer uma colaboração com outra instituição. Nesse sentido, é importante que o marégrafo seja modular, de fácil transporte, instalação e manutenção, e autossustentável em relação ao fornecimento de energia elétrica e armazenamento de dados. Consideramos assim que o projeto não deveria depender de uma tomada local de energia elétrica e, tão pouco, de um acesso à rede internet via conexão wi-fi para o envio dos dados.

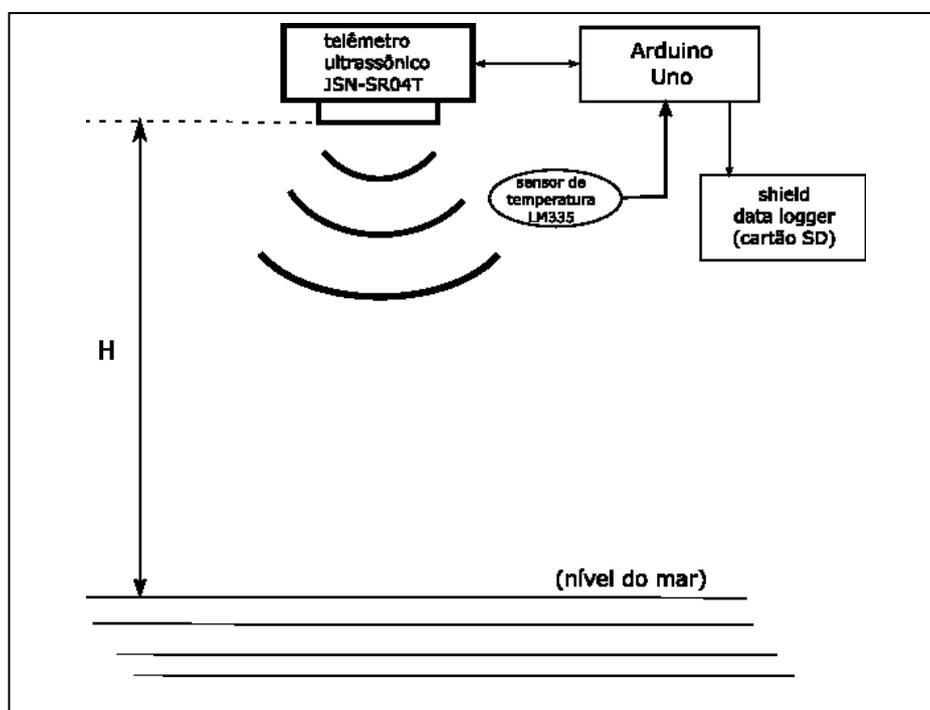


Fig. 1 – Representação esquemática do marégrafo ultrassônico. O telêmetro ultrassônico, modelo JSN-SR04T-2.0 (EC Buying, China), mantido em uma posição fixa, detecta a posição da superfície do mar medindo a distância H . Essa distância é determinada de forma indireta através da medida do tempo gasto por um trem de pulsos ultrassônicos (40 kHz) entre emissão e recepção do eco. A medida da temperatura é feita através do sensor LM335 (TI, USA). Uma placa Arduino Uno R3 e um escudo (shield) datalogger controlam e armazenam os dados em cartão de memória SD.

O esquema básico do projeto pode ser resumido na Fig. 1. Um telêmetro mantido em uma posição fixa em relação ao piso oceânico, a uma altura H em relação ao nível médio do

mar, emite um trem de pulsos ultrassônicos na direção vertical, de cima para baixo. Os pulsos são refletidos na superfície do mar e são captados pelo telêmetro, que mede assim o intervalo de tempo entre a emissão e a recepção dos pulsos (Δt). A metade desse tempo multiplicado pela velocidade de propagação do som no ar livre (v_{som}) fornece a distância H (altura) percorrida. Essa distância serve como referência do nível do mar, ou intensidade da maré. Quanto menor essa distância, mais alto é o nível do mar em relação ao piso oceânico, e quanto maior essa distância, mais baixo é o nível do mar.

A altura H é dada pela relação,

$$H = v_{\text{som}} \cdot (\Delta t/2), \quad (1)$$

e a velocidade de propagação do som, por sua vez, depende da temperatura do ar através da relação,

$$v_{\text{som}} = \sqrt{\frac{\gamma RT}{M}} \quad (2)$$

onde R é a constante universal dos gases, γ é a razão dos calores específicos molares, M é a massa molecular e T a temperatura absoluta do ar (RESNICK, R.; HALLIDAY, D.; KRANE, K. S.). Tomando por base a velocidade de propagação do som no ar a 20 °C de 343 m/s podemos reescrever a relação (2) como,

$$v_{\text{som}}(T) = 20,04 \cdot \sqrt{T} \quad (m/s) \quad (3)$$

Diferente do modelo HC-SR04 que o antecedeu, este bem conhecido e com várias aplicações em atividades de ensino, o modelo JSN-SR04T-2.0 possui um único sensor que atua como emissor e receptor de pulsos ultrassônicos (Fig. 2). O sensor é encapsulado o que permite a sua aplicação em ambientes externos.

A alimentação de 5V é feita através da placa Arduino e os pinos *Trigger* e *Echo* são controlados através das portas digitais D5 e D6 da placa, respectivamente.

Um sensor de temperatura LM335 (-40°C – 100°C, acurácia $\pm 1^\circ\text{C}$) da *Texas Instruments* (USA), com encapsulamento TO-92, é utilizado para monitorar a temperatura do ar. Na Fig. 3 mostramos o sensor e o esquema de ligação. Na montagem atual do marégrafo, optamos por não usar o pino ADJ (*adjust*, terceiro pino à esquerda do sensor – ver Fig. 3) que permite a calibração local do sensor através do ajuste de um potenciômetro. Aproveitando a alta linearidade de resposta deste sensor, optamos por corrigir a temperatura via software. A tabela de dados (*datasheet*) do LM335 e dos demais sensores utilizados podem ser facilmente encontrados na internet.

A temperatura é usada para corrigir a velocidade de propagação do som. Como veremos mais adiante, no software de controle um primeiro registro de altura H é feito tomando por base a velocidade do som como fixa e igual a 344 m/s, a qual corresponde a uma temperatura ambiente de aproximadamente 22°C. É somente na etapa de análise dos dados que as alturas são recalculadas para a temperatura do momento. Considerando-se a amplitude térmica alcançada diuturnamente e, considerando-se, um registro contínuo de vários dias, essa

normalização com a temperatura atual é importante.

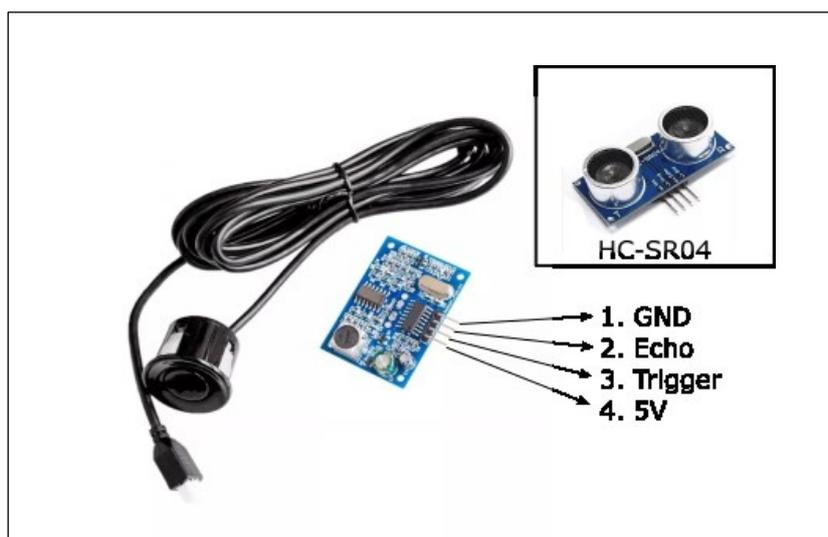


Fig. 2 – Telêmetro JSN-SR04T-2.0, com a emissão e recepção de sinais ultrassônicos em uma única unidade encapsulada, o que possibilita a sua utilização em ambientes externos. A pequena foto mostra o modelo antecessor HC-SR04, mais difundido, baseado em dois sensores independentes, um atuando como emissor e o outro como receptor. Os dois modelos são completamente compatíveis e podem ser facilmente intercambiados. No marégrafo em discussão, a alimentação de 5V e GND (ground) é suprida pela própria placa Arduino.

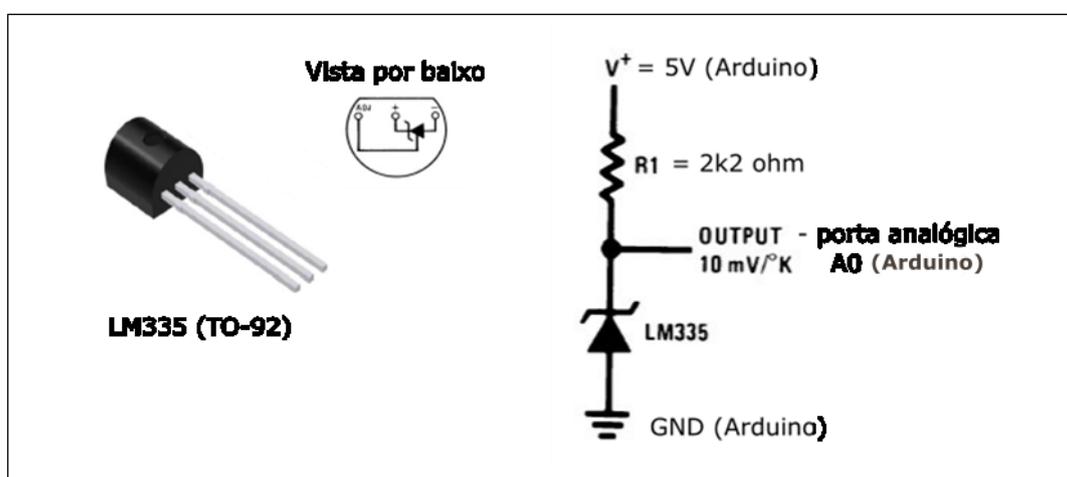


Fig. 3 – Sensor de temperatura LM335 e esquema de ligação.

O escudo (*shield*) *data logger* para Arduino Uno utilizado neste projeto é um produto genérico que segue o modelo criado originalmente pela empresa Adafruit (USA). Este modelo se caracteriza por conter um relógio próprio que disponibiliza a data (dia-mês-ano) e o tempo (hora-minuto-segundo) no momento do armazenamento de uma dada grandeza física que

esteja sendo monitorada. A unidade que executa esta função de relógio é o RTC DS1307. RTC é o acrônimo de *Real Time Clock* e DS1307 é código do circuito integrado, fabricado pela empresa Maxim Integrate (USA), que executa essa função. O relógio, de muito baixo consumo, é mantido por uma pequena bateria de 3V fixada no próprio *shield* e que garante o funcionamento contínuo por mais de um ano. A unidade de armazenamento é um cartão de memória SD formatado em padrão FAT16 ou FAT32. Na Fig. 4 mostramos este *shield data logger*.



Fig. 4 – Escudo data logger para Arduino Uno. Na figura vemos o slot para cartão SD e o suporte para bateria modelo CR1220 de 3V. O escudo possui uma pequena área para prototipagem de grande utilidade.

O circuito básico do marégrafo pode ser visto na Fig. 5. Nesta figura, fazemos uma representação do circuito com auxílio do programa Fritzing (acesso gratuito), onde os componentes são representados com sua forma real. No desenho mostramos a visão de cima para baixo com o escudo *data logger* encaixado sobre a placa Arduino, ou seja, pinos macho do escudo se encaixam exatamente sobre a pinagem fêmea da placa Arduino que está abaixo e, portanto, não visível no desenho.

III. O Watchdog

O circuito de controle do marégrafo mostrado na Fig. 5 foi programado para coleta de dados em períodos de 1/2 minuto durante intervalos, tipicamente, de uma ou duas semanas, que são armazenados em um cartão de memória SD. Um dos problemas deste método é a questão do travamento do microcontrolador: em períodos longos de coleta sequencial de dados de um sensor, é possível que ocorram travamentos da unidade de processamento, fato bastante conhecido por todos os projetistas de sistema de automação do controle de processos e coleta de dados. É bastante frustrante retornar ao marégrafo após uma semana de espera, e verificar que está travado com uma coleta de dados que se resumiu às primeiras horas de funcionamento!

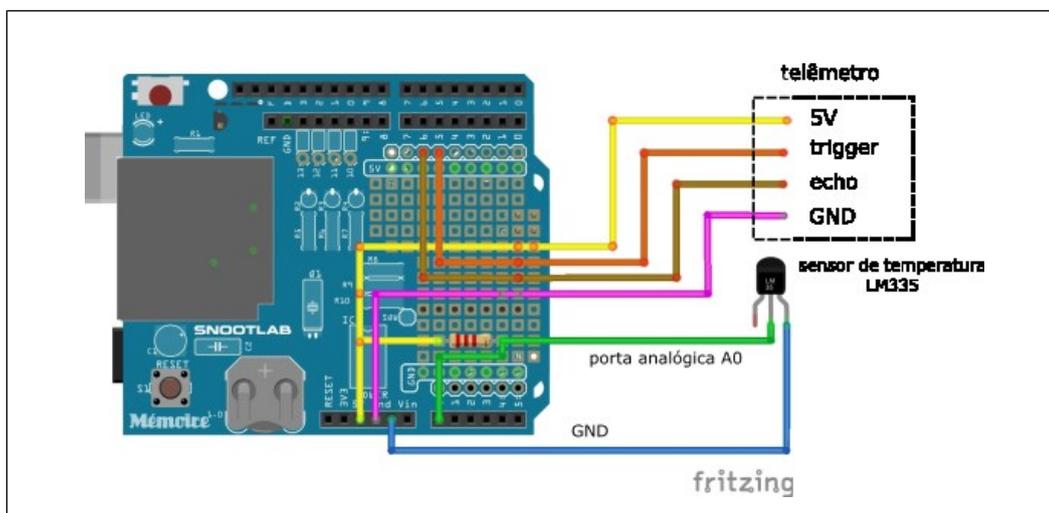


Fig. 5 – Circuito eletrônico básico do marégrafo desenhando com auxílio do programa Fritzing. O desenho mostra o escudo data logger com a placa Arduino abaixo. Mostramos também as conexões do telêmetro JSN-SR04T-2.0 e do sensor de temperatura LM335.

Para atenuar os problemas de travamento foram desenvolvidos vários recursos de hardware e software capazes de detectarem a interrupção de execução do programa e agir de modo a reiniciá-lo. Um tal recurso é conhecido, genericamente, como “cão vigia”, ou em sua versão em inglês, *watchdog*. Existem vários tipos, mas nos parece que as soluções mais simples e confiáveis são aquelas constituídas de circuitos eletrônicos externos ao circuito monitorado, ou seja, soluções via hardware autônomo, externo ao circuito principal. Neste projeto fizemos uso de um *watchdog* baseado no circuito integrado 555 (RAMSAY, C.). Sobre o CI555 especificamente existem também muitas referências em livros e na internet e, portanto, damos aqui uma descrição simples e focada na aplicação. Na Fig. 6 mostramos o circuito do *watchdog* baseado no CI 555 utilizado em modo astável. O desenho do circuito, em formato convencional, foi feito também com auxílio do programa Fritzing. A forma como este circuito pode ser construído e aplicado em cada projeto é bem flexível. Na versão atual do projeto, o *watchdog* foi montado sobre um mini *proto-board* de 170 pinos, como mostrado também na Fig. 6.

Quando a placa Arduino é ligada, o capacitor C2 começa a carregar através dos resistores R1 e R2. O tempo efetivo de carga de C2 neste circuito é dado por,

$$t_{carga} = 0,693(R1 + R2)C2$$

Para os valores indicados na figura, $t_{carga} \approx 70$ s (ou 1 min 10 s). Esse tempo, como veremos, marca o intervalo de observação: se o sistema travar, 70s após o travamento o *watchdog* reinicializa a Arduino. Quando a tensão no pino 6 (pino *trigger*) atinge o valor de 2/3 da tensão de alimentação (5V) o pino 7 (pino *discharge*) é levado ao estado baixo e o capacitor C2 descarrega através da resistência R2.

O tempo de descarga é dado por,

$$t_{descarga} = 0,693 \cdot R2 \cdot C2$$

que para os valores indicados na Fig. 6 é de 0,32 s. Com a descarga de C2 todo o processo se reinicia.

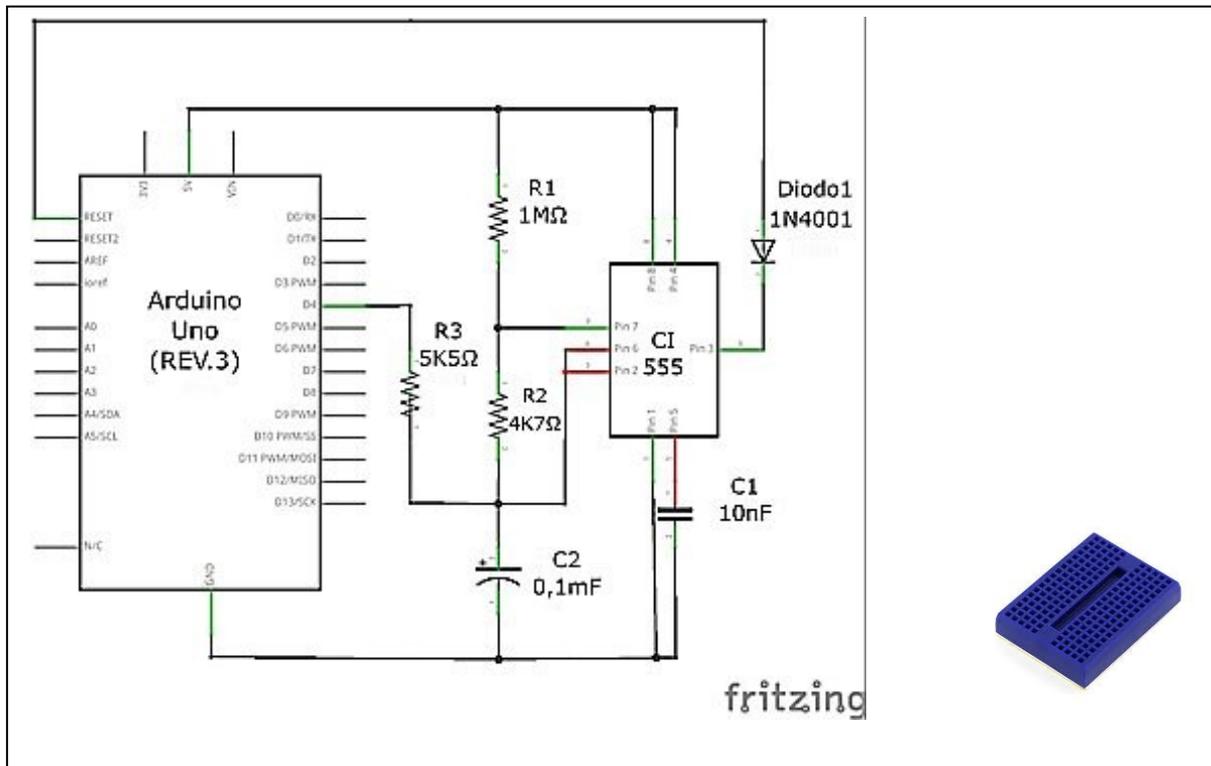


Fig. 6 – Circuito do watchdog externo baseado no CI 555 em modo astável (esquerda). Na versão atual do projeto, o watchdog foi implementado num pequeno protoboard de 170 pinos e fixado lateralmente à placa Arduino (direita). A conexão com a Arduino é feita através de fios jumpers.

Como vemos na Fig. 6, o pino 3 (*output*) do 555 está ligado ao pino de RESET da placa Arduino através do Diodo 1. Tanto o pino RESET quanto o próprio botão RESET podem ser usados para a reinicialização da placa Arduino. O pino de RESET fica normalmente no estado alto (HIGH), e se for levado ao estado baixo (LOW) por uma ação externa, o programa é reinicializado. Assim, na sequência do que discutimos acima, toda vez em que se completar a descarga de C2, a placa Arduino será reinicializada.

Temos agora os elementos para entender como funciona o *watchdog*: para evitar que a placa Arduino seja reinicializada continuamente temos que criar no programa em execução um evento que evite que a tensão de carga de C2 chegue a $2/3$ V (pino 6)! Esse truque é feito através do pino digital D4 da Arduino. Se colocamos o pino D4 no estado LOW, forçamos a descarga do capacitor C2 através do resistor R3. Essa descarga pode ser rápida e periódica de

forma a manter a tensão no pino 6 (TRIGGER) do 555 abaixo de $2/3 V$. Se ocorrer um travamento na execução do programa, a estância em que D4 é colocado em estado LOW não é executada e, portanto, a tensão de carga segue normalmente até atingir $2/3 V$, ocorrendo, com isso, a reinicialização. No Apêndice, listamos o *sketch* desenvolvido para o controle do marégrafo.

IV. Um exemplo de montagem do marégrafo ultrassônico

O marégrafo pode ser montado de formas diferentes, de maneira que atenda as especificidades de cada local de observação. A montagem que apresentamos foi concebida para instalação num píer de atracamento que fica no Centro de Instrução Almirante Graça Aranha (CIAGA) da Marinha do Brasil, na Avenida Brasil, no. 9020, Olaria, na cidade do Rio de Janeiro. Esse ponto de observação fica bem no interior da Baía de Guanabara tendo em frente o Aeroporto Internacional Tom Jobim, na Ilha do Governador.

A estrutura do marégrafo foi concebida tendo por base tubulações de PVC usadas em instalações hidráulicas, do tipo soldável (PVC marrom), de 32 mm. A montagem pode ser melhor apreciada através das fotografias apresentadas na Fig. 7. Na Fig. 7(a) vemos o conjunto completo. As hastes de sustentação do painel solar e do sensor ultrassônico são conectadas à base por uniões de PVC para assim permitir um giro axial. Essa mobilidade é importante para poder orientar o painel solar na direção azimutal mais favorável (norte) e orientar o eixo do sensor ultrassônico na direção perpendicular à lâmina d'água. O painel solar é preso à haste vertical por abraçadeiras, o que permite a orientação zenital mais favorável para o período de trabalho. A caixa-abrigo para os circuitos eletrônicos (placa Arduino com acessórios e circuito controlador de carga do painel fotovoltaico) é do tipo usado em quadro de disjuntores externo em instalações elétricas. Trata-se de uma caixa de plástico resistente ao sol e à chuva, com porta de acesso transparente e com isolamento de água. Existem diferentes modelos que podem ser selecionados para diferentes necessidades, mas deve-se estar atento à eficácia do isolamento de água, por conta das chuvas e da alta insolação. Esta caixa deve ser adequadamente ventilada e, se necessário, alguns furos podem ser feitos para facilitar a circulação de ar. No nosso caso, ela foi colocada bem abaixo do painel solar para ficar sob a sombra e abrigada da chuva.

Uma base de madeira é assentada sobre a estrutura com dois parafusos e buchas plásticas, para a fixação do marégrafo no piso do píer. Sobre a madeira fixamos a bateria com abraçadeiras de nylon, e sobre a bateria, instalamos uma caixa plástica coberta com papel alumínio. Na Fig. 7(c) vemos o abrigo do sensor ultrassônico, que consiste de um tampão de tubulação de 100 mm em PVC branco, utilizado em instalações sanitárias. No centro do tampão fixamos uma luva de 32 mm no interior da qual se adapta o sensor ultrassônico. Esse abrigo é fixado lateralmente a uma forquilha, construída com barra chata de alumínio de 1 polegada/3mm, através de dois parafusos com porcas borboletas. Essa forma de fixação permite girar o abrigo e completar a orientação do sensor perpendicular à lâmina d'águ

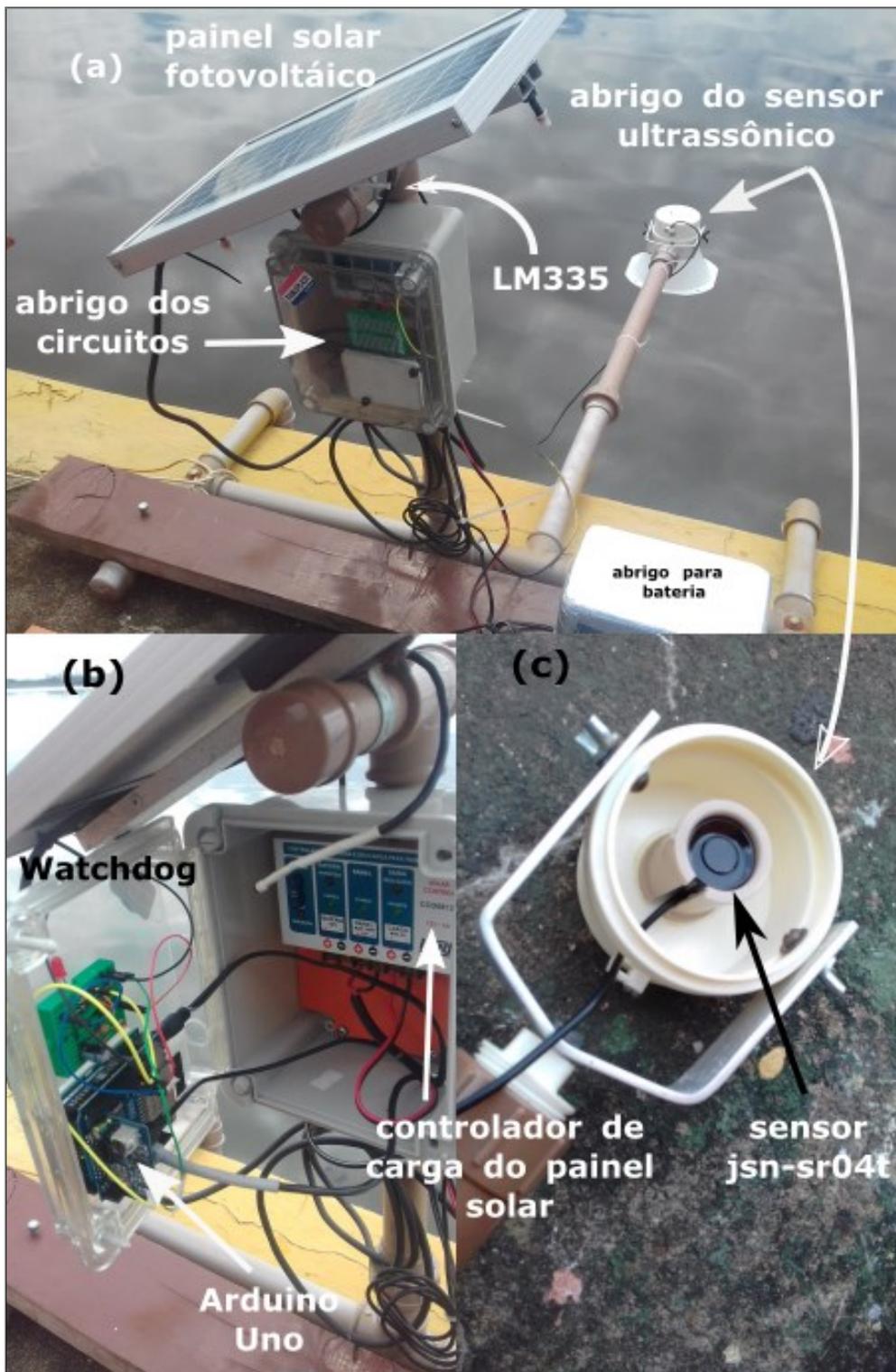


Fig. 7 – Montagem do marégrafo ultrassônico no Centro de Instrução Almirante Graça Aranha.

A Fig. 7(b) mostra o interior da caixa-abrigo onde vemos o controlador de carga do painel solar fotovoltaico.

A tampa transparente se tornou uma necessidade prática para facilitar a visualização do circuito de controle e do controlador de carga do painel solar. Antes de se retirar da estação maregráfica, é necessário que o operador verifique se todo o sistema está funcionando adequadamente. Certificamo-nos do bom funcionamento observando se os LED's de aviso estão acendendo corretamente. Por fim, a colocação do cone corresponde a uma tentativa de colimar um pouco mais a saída dos pulsos ultrassônicos. Na captação, o cone atua produzindo a convergência do sinal de eco, e a eficácia desse item está ainda em fase de avaliação.

Para a alimentação, usamos uma bateria de 12V/7Ah de uso comum em *nobreaks*, uma placa solar fotovoltaica Kyocera, modelo Ks5t e um controlador de carga e descarga para painel solar da Lead, modelo CCD0812 de 12V/8A. Um texto contendo mais detalhes desta montagem pode ser encontrado em

www.if.ufrj.br/~pef/producao_academica/dissertacoes/2019_Rafael_Soares/artigo.

Disponibilizamos também um conjunto de fotos e vídeos avulsos que retratam a montagem e o local de instalação.

V. Resultados

Na Fig. 8 apresentamos alguns resultados coletados. Podemos ver uma comparação de dados observados no marégrafo ultrassônico instalado no CIAGA e os dados coletados, para o mesmo período, no marégrafo instalado na base da Ponta da Armação do Centro de Hidrografia da Marinha (CHM). A estação do CHM fica mais próxima da entrada da Baía da Guanabara, na cidade de Niterói/RJ. O marégrafo da Ponta da Armação é do tipo radar, baseado na diferença de tempo entre a emissão e recepção de sinais eletromagnéticas na faixa de micro-ondas. Estes são mais precisos, têm maior alcance e as medidas não são afetadas pela temperatura do ar.

Podemos observar uma boa correspondência entre as duas séries de medida. Para efeito de comparação com os dados do CHM, os dados obtidos no CIAGA foram convertidos de 'distância do nível do mar em relação ao sensor' para 'altura do nível do mar em relação ao piso do mar no local'. Como vemos no gráfico, as alturas medidas nas duas estações são diferentes, pois as referências de 'nível zero' são diferentes para as duas estações. No gráfico indicamos o início das fases da lua para o período de observação.

Podemos ver no registro do marégrafo ultrassônico alguns pontos dispersos. Esta dispersão tende a se concentrar nos períodos de baixamar, isto é, nas menores profundidades, que correspondem à maior distância entre o sensor ultrassônico e a superfície do mar. A região nominal de detectabilidade do sensor JSN-SRT04 está na faixa entre 20 cm e 600 cm, mas acreditamos que o limite superior efetivo, neste caso, seja algo menor, porque uma superfície líquida tende a ter um *coeficiente de reflexão* para ondas sonoras menor do que o de superfícies sólidas, e além disso, devido às ondulações naturais, a superfície do mar não forma um plano liso para a reflexão das ondas ultrassônicas. Para compensar, mantivemos a taxa de amostragem em duas medidas por minuto, o que contrasta com o sensor tipo radar do

CHM que é de uma medida para cada cinco minutos.

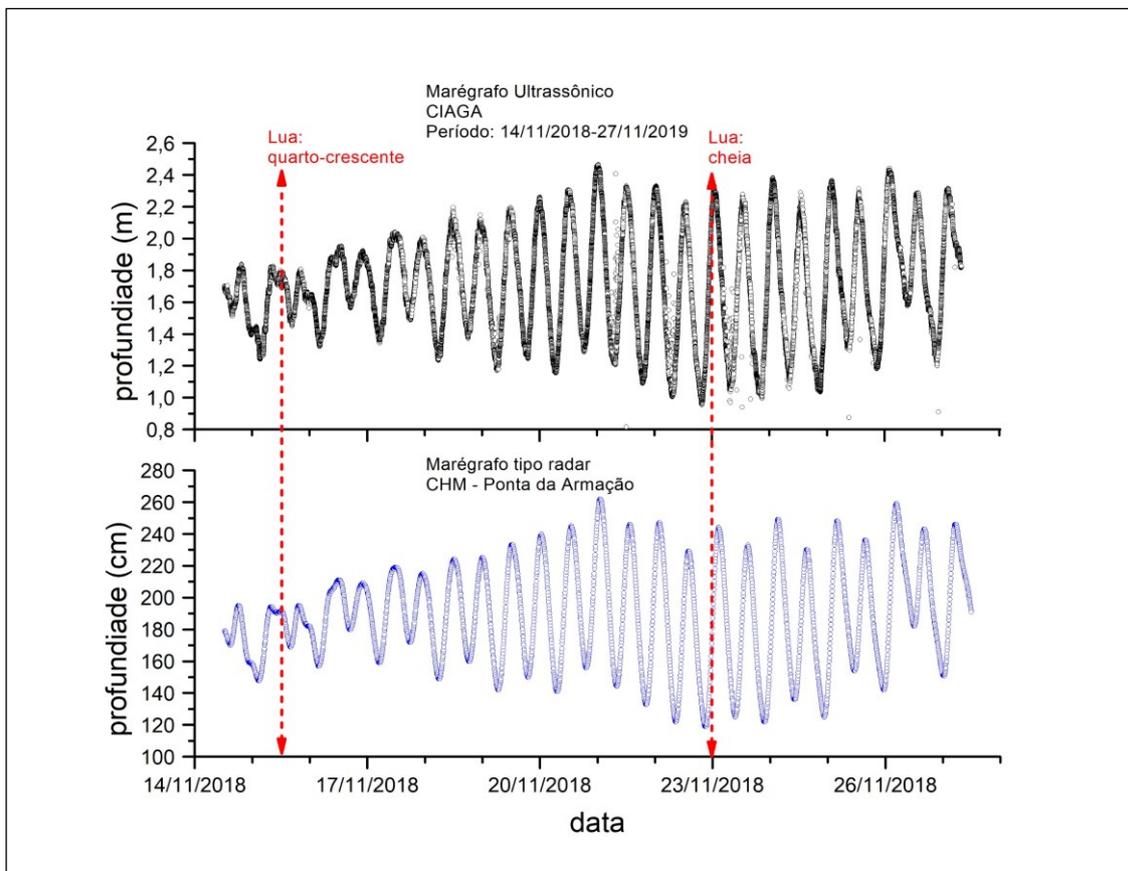


Fig. 8 – Registro mareográfico observado com o marégrafo ultrassônico instalado no CIAGA e o marégrafo tipo radar, do Centro de Hidrografia da Marinha (CHM), instalado na Ponta da Armação, Niterói/RJ. As linhas verticais indicam o início das fases lunares ao longo do período de observação que se estendeu de 14/11/2018 até 27/11/2018. As profundidades indicadas não podem ser comparadas diretamente, pois a referência de nível zero é diferente para as duas estações. O importante a ressaltar é a perfeita compatibilidade dos perfis de linha.

VI. Conclusão

O custo do projeto do marégrafo ultrassônico que apresentamos foi mantido a valores menores do que R\$ 350,00, muito inferior a equipamentos similares disponíveis no comércio especializado. A construção não é exigente e pode ser integralmente feita numa oficina simples. A execução da parte eletroeletrônica é também muito simples e exige tão somente um conhecimento básico de utilização da placa Arduino.

Quanto às aplicações educacionais, são inúmeras, e esperamos ter oportunidade de apresentar algumas sugestões no próximo trabalho. Vemos o projeto como uma boa

plataforma para atividades de investigação. Se por um lado o modelo newtoniano do fenômeno de maré, também referido como *maré astronômica* ou *maré estacionária*, consegue explicar aspectos básicos dos registros maregráficos observados na Fig. 8, ele falha na previsão do horário em que ocorre os extremos da maré, o horário da preamar e do baixamar, e falha na previsão da intensidade em que se manifesta. O modelo newtoniano de maré prevê duas altas e duas baixas de maré no ciclo diurno de 24h, mas os oceanógrafos revelam um quadro bem mais complexo nos oceanos de nosso planeta (GARRISON; CALVO; GIBRAT). São vários aspectos da fenomenologia local abertos à investigação.

Para professores que se interessem em explorar o fenômeno de marés com seus alunos e não têm disponibilidade de acesso a um ponto de observação na costa, como é o caso nas cidades do interior do país, advertimos que é possível ter acesso a dados das estações de observação de maré administrados pelo Centro de Hidrografia da Marinha. São inúmeras as estações instaladas no litoral brasileiro e o contato pode ser feito através da página do CHM na internet. Lá os interessados vão encontrar um serviço muito profissional e muito amigável.

Para finalizar, devemos acrescentar que o projeto do marégrafo ultrassônico não se justifica apenas pelos dados. Queremos lembrar a importância da prática experimental no ensino de Física. A coleta dos dados envolve várias etapas que precisam ser bem planejadas. Para uma análise mais completa do fenômeno de maré, é natural prever a coleta contínua de dados durante o período de, pelo menos uma luação, cobrindo assim as quatro fases da Lua. A investigação pode ser estendida ao longo do ano para a observação da maré nos equinócios e solstícios, onde encontramos diferentes arranjos do sistema Terra-Lua-Sol influenciando as marés de forma diversa. Isso exige trabalho em grupo, avaliação crítica dos resultados quanto à precisão e acurácia final, cuidados de manutenção técnica do equipamento, ou seja, uma série de atividades que são típicas da atividade dos cientistas, do mundo real da Ciência, e que podem fazer da escola uma produtora de conhecimento.

Agradecimentos

Os autores são imensamente gratos ao Centro de Instrução Almirante Graça Aranha pela inestimável ajuda na realização deste projeto. Particularmente gostaríamos de agradecer aos Comandantes José Menezes Filho e Rodrigo Reis Bittencourt, e aos Suboficiais Paulo Sérgio Lopes Sirodio e Francisco Liberato por todo apoio em viabilizarem a instalação do marégrafo e às constantes visitas que fizemos ao CIAGA. Agradecemos também ao Comandante Cesar Henrique de Oliveira Borba, encarregado da Seção de Marés da Diretoria de Hidrografia e Navegação (DHN) – Marinha do Brasil – pelo fornecimento de dados de maré e por sua tão atenciosa acolhida na visita que fizemos ao posto de observação de marés em Ponta da Armação. O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES).

Referências

CALVO, L. L. **Introducción a la Geofísica**. Madrid: Paraninfo, 1972.

CHM - CENTRO DE HIDROGRAFIA DA MARINHA. Disponível em: <<https://www.marinha.mil.br/chm/dados-do-segnav/dados-de-mare-mapa/>>. Acesso em: 01 mai. 2019.

CARVALHO, L. R. M.; AMORIM, H. S. Observando as marés atmosféricas: Uma aplicação da placa Arduino com sensores de pressão barométrica e temperatura. **Revista Brasileira de Ensino de Física**, v. 36, n. 3, p. 3501, 2014.

FRITZING. Disponível em: <<http://fritzing.org/home/>>. Acesso em: 20 jan. 2019.

GARRISON, T. **Fundamentos de Oceanografia**. National Geographic Learning. CENGAG Learning, 2017.

GIBRAT, R. **La Energía de las Mareas**. Nueva Colección Labor, Barcelona, 1973.

RAMSAY, C. Giving an Arduino a Heartbeat. Disponível em: <<https://chrisramsay.co.uk/posts/2015/04/giving-an-arduino-a-heartbeat/>>. Acesso em: 04 mai. 2019.

RESNICK, R.; HALLIDAY, D.; KRANE, K. S. **Física 2**. 5. ed. Ed. LTC, 2003. cap.19. p.149-150.

SILVEIRA, F. L. Marés, Fases Principais da Lua e Bebês. **Caderno Brasileiro de Ensino de Física**, v. 20, n. 1, p. 10-29, 2003.

Apêndice – O programa (*sketch*)

Na sequência, transcrevemos a última versão do *sketch* de controle e aquisição de dados:

/*

Programa para o controle de um telêmetro ultrassônico e armazenamento de dados de maré em cartão de memória tipo SD - Versão 4.1 de 18/02/2019

Nesta versão 4.0 usamos um cartão datalogger que inclui um relógio (RTC- Real

Time Clock) e implementamos um *watchdog* externo. Incluímos a medida da temperatura da estação maregráfica com auxílio de um sensor LM335. Um fator de calibração é incluído para correção das medidas de temperatura via software.

O *watchdog* tem uma janela atual de espera de 70s. Qualquer alteração tem que ser feita através da troca do resistor e/ou do capacitor de carga.

Nesta versão 4.1 mudamos o formato de gravação dos dados para que seja compatível com o usado pelo CHM (18/02/2019).

O presente programa faz uso de contribuições de inúmeros autores, sendo difícil uma referência completa de todos. Assim sendo, os autores agradecem a grande comunidade de desenvolvedores de projetos para a plataforma Arduino que se caracteriza de forma notável pelo desprendimento, cooperação e generosidade.

Trabalho de tese do Prof. Rafael Guedes Soares.

Composição do circuito:

- * Telêmetro ultrassônico jsn-sr04t
- * Data Logging shield v1.0 - www.deek-robot.com
- * Sensor de temperatura LM335
- * *watchdog* baseado no CI 555.
- */

```
#include <SPI.h>
```

```
#include <SD.h>
```

```
#include <Wire.h>
```

```
#include "RTCLib.h"
```

```
#define LOG_INTERVAL 30000 // estabelece período de 30 segundos
```

```
#define echoPin 6 //Pino 6 recebe o pulso do echo
```

```
#define trigPin 5 //Pino 5 envia o sinal para gerar o pulso
```

```
float altura = 0;
```

```
long tempo = 0;
```

```
int Kelvin = 0; //temperatura ambiente em Kelvin
```

```
int dK = -3; // calibração da temperatura (K) em 09/04/2019
```

```

// pinos usados para conectar os led's do data logger shield
#define redLEDPin 3
#define greenLEDPin 2

#define pinoDreno 4 // pino de dreno do watchdog
#define pinoKelvin 0 // pino analógico usado para o LM335

RTC_DS1307 RTC; // define um objeto do Real Time Clock

// pino 10 é usado pelo escudo data logging para gravação do cartão de memória
const int chipSelect = 10;

File logfile; // arquivo de armazenamento

void error(){
  //indicador de erro através do led vermelho do data logger
  for(uint8_t i = 0; i < 10; i++)
    digitalWrite(redLEDPin, HIGH);
  delay (1000);
  digitalWrite(redLEDPin, LOW);
}

void watchdog(){
  /*Essa função, quando chamada, descarrega o capacitor C2.
  Sendo chamada não permite que o pino reset seja acionado. Ocorrendo um travamento, a
  execução do programa é interrompido a função não é acionado ocasionando o RESET*/
  pinMode(pinoDreno, OUTPUT);
  digitalWrite(pinoDreno, LOW);
  delay (300);
  //retorna o pino para alta impedância
  digitalWrite(pinoDreno, HIGH);
}

void setup(){

  //Ativa watchdog
  watchdog();

  // LEDs de controle (debugging)

```

```

pinMode(redLEDpin, OUTPUT);
pinMode(greenLEDpin, OUTPUT);

pinMode(chipSelect, OUTPUT);
pinMode(echoPin, INPUT); // define o pino 6 como entrada (recebe)
pinMode(trigPin, OUTPUT); // define o pino 5 como saída (envia)

// verifica se o cartão de memória pode ser inicializado
if (!SD.begin(chipSelect)) {
    error();
}

// cria arquivo novo
char filename[] = "LOGGER00.dat";
for (uint8_t i = 0; i < 100; i++) {
    filename[6] = i/10 + '0';
    filename[7] = i%10 + '0';
    if (!SD.exists(filename)) {
        // verifica se o arquivo já existe,
        //o arquivo é aberto somente se não existe no cartão de memória
        logfile = SD.open(filename, FILE_WRITE);
        break; // sai do loop.
    }
}

if (!logfile) {
    error();
}

// conecta o RTC.
Wire.begin();
RTC.begin();
logfile.println("Marégrafo - Versão 4.1");
logfile.println("data/HH:mm:ss;  Altura (cm);  Temperatura(K);");
}

void loop(){
    DateTime now;

```

```

// intervalo entre leituras.
delay(LOG_INTERVAL);

digitalWrite(greenLEDpin, HIGH);

// registra o tempo atual.
now = RTC.now();

logfile.print(now.day(), DEC);
logfile.print("/");
logfile.print(now.month(), DEC);
logfile.print("/");
logfile.print(now.year(), DEC);
logfile.print(" ");
logfile.print(now.hour(), DEC);
logfile.print(":");
logfile.print(now.minute(), DEC);
logfile.print(":");
logfile.print(now.second(), DEC);
logfile.print(";");

digitalWrite(trigPin, LOW); //seta o pino 5 com um pulso baixo "LOW" ou desligado ou
ainda 0
delayMicroseconds(5); // delay de 2 microssegundos
digitalWrite(trigPin, HIGH); //seta o pino 5 com pulso alto "HIGH" ou ligado ou ainda 1
delayMicroseconds(15); //delay de 12 microssegundos
digitalWrite(trigPin, LOW); //seta o pino 5 com pulso baixo novamente
long duration = pulseIn(echoPin,HIGH); //pulseIn lê o tempo entre a chamada e o pino
entrar em high

//Esse cálculo é baseado em  $s = v \cdot t$ , lembrando que o tempo vem dobrado
//pois corresponde ao tempo de ida e volta do pulso. Velocidade do som = 344 m/s.

altura = float(duration*0.0172);
Kelvin = analogRead(pinoKelvin);
Kelvin = int(Kelvin*5/10.23) - dK;

logfile.print(altura,1);
logfile.print(";");

```

```

logfile.print(Kelvin);
logfile.println(";");
digitalWrite(greenLEDpin, LOW);

// led pisca indicando a gravação dos dados
digitalWrite(redLEDpin, HIGH);
logfile.flush();
digitalWrite(redLEDpin, LOW);

watchdog();

}

```

A variável dK deve ser determinada, no início dos trabalhos de coleta de dados, por leitura direta da temperatura ambiente com auxílio de um termômetro calibrado. Devemos medir a temperatura registrada no sensor LM335 e determinar a discrepância com o valor obtido com o termômetro calibrado,

$$dK = (\text{Temperatura do termômetro calibrado}) - (\text{Temperatura lida no LM335})$$

A listagem a seguir mostra uma típica saída de dados:

Marégrafo - Versão 4.1

data/HH:mm:ss; Altura (cm); Temperatura(K);

16/4/2019 11:11:19;162.0;307;

16/4/2019 11:11:49;162.0;304;

16/4/2019 11:12:19;161.6;305;

16/4/2019 11:12:50;161.6;305;

.....

A constante LOG_INTERVAL determina a periodicidade em que as medidas são realizadas. Usamos o valor de 30000 ms, ou 30 s, o que é, de fato, um valor muito baixo para a maioria dos estudos que se deseja fazer sobre o comportamento das marés. Um valor adequado de 5 minutos já seria bastante adequado. Entretanto, como há uma quantidade imprevisível de falhas, ou dados errados, uma frequência maior na coleta dos dados pode compensar essas falhas.



Direito autoral e licença de uso: Este artigo está licenciado sob uma [Licença Creative Commons](https://creativecommons.org/licenses/by-nc-nd/4.0/).