Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Bacharelado em Ciência da computação

Implementação de um sistema de controle para uma estação de sensoriamento sem fio aquática

Bruno Marques Cremonezi

JUIZ DE FORA
JUNHO, 2014

Implementação de um sistema de controle para uma estação de sensoriamento sem fio aquática

Bruno Marques Cremonezi

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Bacharelado em Ciência da computação

Orientador: Alex Borges Vieira

JUIZ DE FORA

JUNHO, 2014

Implementação de um sistema de controle para uma estação de sensoriamento sem fio aquática

Bruno Marques Cremonezi

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Alex Borges Vieira
Doutor

Luciano Jerez Chavez
Mestre

Francisco Henrique Cerdeira Ferreira
Mestre

JUIZ DE FORA

26 DE JUNHO, 2014

Resumo

Redes de Sensores Aquáticos (RSA) é uma área em ascensão, tanto no meio acadêmico quanto no industrial. Nota-se hoje no Brasil uma grande necessidade de tecnologias avançadas para o monitoramento de seus recursos hídricos, já que o país possui um território privilegiado por tais recursos. Neste trabalho será apresentado um *software* de controle para o nó sensor Hydronode, uma sonda limnológica de última geração com tecnologia eletrônica, *software* e parte mecânica totalmente projetada e desenvolvida no Brasil. Desenvolveu-se também todas as funcionalidades para o controle da rede, bem como todos módulos de transformação e visualização dos dados recebidos do nó sensor. Buscamos minimizar a dependência tecnológica das pesquisas brasileiras em plataformas estrangeiras de RSA, criando uma plataforma brasileira de sistemas de monitoramento robusta com capacidade para competir com tecnologias similares desenvolvidas na Alemanha, Japão ou EUA.

Palavras-chave: Hydronode, RSA, Software, Plataforma.

Agradecimentos

Aos professores, amigos e orientadores Alex Borges Veira e Guilherme Albuquerque Pinto, por acreditarem em meu potencial, oferecendo apoio desde o príncipio.

Aos meus pais, pelo apoio e paciência ao longo de todos esses anos.

Aos meus amigos de curso: Igor Russo, Marcelo Marques, Thiago Marques e Victor Patrocínio, pessoas com que tive o prazer de conviver durante quatro anos da minha vida.

A Mariana de Almeida Nery por me apoiar e incentivar mesmo nos momentos difíceis e por me fazer acreditar em mim mesmo.

Para todos amigos, que mesmo de maneira indireta, foram peças essenciais para conclusão a deste projeto e faculdade, especialmente André Henriques, Camila Mota, Diego Henrique, Daniel de Filippo, Narjara Oliveira e toda equipe Clack.

"Talvez não tenha conseguido fazer o melhor, mas lutei para que o melhor fosse feito. Não sou o que deveria ser, mas Graças a Deus, não sou o que era antes".

Marthin Luther King

Sumário

Lista de Figuras

1 Introdução

Redes de sensores aquáticos (RSA) é uma importante área que vem ganhando interesse no meio acadêmico e industrial. Lagos, rios e oceanos são de grande importância para o equilíbrio da vida na terra. O monitoramento dessas regiões possui aplicações como oceanografia, estudo da interação oceano e atmosfera, biologia marinha, estudo do clima, aquecimento global, arqueologia, predições sísmicas, detecção de poluentes e substâncias contaminantes, controle da qualidade da água e exploração e monitoramento de campos de gás, óleo e petróleo (Vieira e Luiz Filipe Menezes, 2009).

Exemplificando as área de aplicação apresentadas na área de oceanografia, RSAs são utilizadas para analisar e prever variadas características do ambiente. Os dados pelas RSAs podem ser utilizados para diminuir os riscos de uma navegação (Akyildiz et al., 2005). Temos aplicações na área militar também, onde RSAs podem ser utilizadas para detectar e caçar submarinos (Kong et al., 2005), bem como podem auxiliar na localização de minas submarinas. Na área industrial, podemos citar a aplicação de RSA no controle e monitóramento de tubos aquáticos bem como o monitoramento de maquinário de pesca (Heidemann et al., 2012). Por fim podemos citar a utilização de RSAs no setor de energia, em que através desta tecnologia, consegue-se detectar mexilhões ouro e amêijoas da china que infectam barragens das hidrelétricas brasileiras, podendo levar à prejuízos e catástrofes ambientais.

Uma RSA é formada por diversos nós sensores autônomos. Os recentes avanços na tecnologia *wireless* e nos microprocessadores possibilitaram o desenvolvimento de sensores, aparelhos com baixo custo, pouco consumo de energia e multifuncionais (Loureiro et al., 2003; Nayak; Amiya and Ivan Stojmenovi c, 2010). Esses pequenos aparelhos têm a função de capturar fenômenos ocorridos e estimar valores parâmetros em função do evento decorrido.

Redes de Sensores Sem Fio (RSSFs) representam, então, uma evolução natural dos sensores comuns. Uma RSSF é composta por diversos sensores os quais são espalhados diretamente na área do fenômeno que se deseja estudar, ou próximos a ele. Essas

redes tendem a possuir um grande número de nodos, restrições de energia e mecanismos de autoconfiguração, ou seja, problemas como perda de nodos e falha de comunicação são comuns nesse tipo de rede, uma vez que mudanças na topologia da rede são muito frequentes (Akyildiz et al., 2005).

Uma RSSF tem como tendência ser autônoma e cooperativa para executar as tarefas da rede. Ao contrário de sobrecarregar os nodos responsáveis pela fusão de dados, a rede utiliza cada nodo e sua capacidade de processamento localmente, transmitindo apenas dados necessários e parcialmente processados (Loureiro et al., 2003).

Redes de Sensores Aquáticas emergiram rapidamente para diversas aplicações importantes, um nó sensor aquático (NSA) pode detectar características do ambiente, coletar dados e se comunicar com outros nós. Um dos principais desafios da implantação de uma rede de nós sensores aquáticos é o desenvolvimento do nó, incluindo o seu elevado custo quando comparado a nós sensores terrestres. A maioria das arquiteturas de hardware dos nós de uma rede de sensores objetivam aplicações específicas. Pesquisadores necessitam de uma plataforma unificada para testar o desempenho prático de protocolos de rede.

Do ponto de vista acadêmico, poucas plataformas são desenvolvidas para uma RSA. Neste trabalho, vamos dar o foco para a tecnologia Hydronode (Viana et al., 2012), desenvolvida conjuntamente pelas Universidades Federais de Minas Gerais, Viçosa e Juiz de Fora.

O HydroNode é uma plataforma para rede de sensores aquáticos constituída de vários nós. Cada nó possui até oito sensores, escolhidos de acordo com o objetivo de pesquisa, aumentando a flexibilidade da plataforma. Os nós dessa rede se comunicam em ambiente aquático mesmo quando estão submersos, por meio de um modem acústico. O HydroNode tem a capacidade de coletar dados de temperatura da água, oxigênio dissolvido, condutividade, pH, clorofila e turbidez. Esses dados são armazenados em uma base emersa, onde ficam disponveis para a coleta.

Os dados de leitura dos sensores são requisitados pela base emersa e enviados para a sonda central por meio de comunicaçãao serial. A sonda central propaga a mensagem de requisição de leitura por meio de comunicação acústica para os outros nós. Os nós

que recebem a mensagem realizam a leitura dos sensores e retornam os dados lidos para a sonda central, que repassa a mensagem para a base.

A coleta dos dados do HydroNode é feita por um computador pessoal utilizando um cabo de comunicação serial conectado à base emersa. Essa comunicaçãao permite também o envio dos dados de cofiguração da rede.

Neste trabalho é apresentado o *software* de controle que realizará a comunicação com a *Manager Board*, placa destinada a efetuar o gerenciamento de todos os processos do nó, que por sua vez se comunica com a *Acquisition Board*, placa responsável pela aquisição de dados dos sensores presentes no nó. Nessa aplicação o usuário tem a opção de apenas exibir os dados coletados, ou armazená-los em um banco de dados.

2 Motivação e Objetivos

Sensores fazem parte do nosso dia-a-dia em diversas situações e cenários. Sensores de presença, de monitoramento de temperatura e de umidade, de controle de estoque estão espalhados em nosso cotidiano. Na maioria dos casos, esses sensores formam redes nas quais informações são armazenadas, processadas ou simplesmente repassadas entre os sensores que as compõem. Essas redes são denominadas redes de sensores sem fio (RSSFs) e diversos artigos na literatura apresentam aplicações e algoritmos direcionados a essa classe de redes (Akyildiz et al., 2007).

Redes de Sensores Aquáticas é uma importante área de pesquisa que está atraindo um interesse, tanto da comunidade de pesquisa como também da indústria. Oceanos, rios e lagos são fundamentais para a vida em nosso planeta e o monitoramento desses ambientes é uma tarefa difícil e dispendiosa. Assim, existe um grande número de aplicações em que RSAs são importantes como a preservação do ecossistema, prevenção e monitoração de vazamento de óleo/gás, exploração e gestão de reservatórios de água doce.

Nos últimos anos, o interesse pelo monitoramento de regiões subaquáticas cresceu expressivamente. Países monitoram reservatórios de exploração de petróleo, buscando algum vazamento; usinas hidrelétricas monitoram constantemente o nível de água, buscando escapamentos ou mesmo ajudando a estabelecer políticas de racionamento que eventualmente possam ocorrer; biólogos procuram estudar a vida marinha, monitorando espécies existentes ou até mesmo catalogando uma nova espécie (Chandrasekhar et al., 2006). Dessa forma, com implementação de um monitoramento subaquático é possível estabelecer um constante monitoramento das condições submarinas, em tempo real, a fim de obter informações importantes do ecossistema.

Porém, o monitoramento desse tipo de região apresenta diversos desafios. É possível citar, por exemplo, a movimentação dos sensores pelas correntes aquáticas e a preocupação de estender ao máximo a vida da bateria, visando aumentar a vida da rede como um todo.

Nota-se, hoje no Brasil, uma grande necessidade de tecnologias avançadas para o monitoramento de seus recursos hídricos, já que o país possui um território privilegiado em potencial hídrico, contendo umas das maiores reservas de água doce do mundo, sendo essas reservas distribuídas entre rios, lagos, manguezais, açudes, lagos, entre outros.

No Brasil, não há uma tradição de trabalho em redes de sensores aquáticas. Equipamentos e *softwares* são importados, assim há uma grande dependência em relação a um conjunto muito restrito de indústrias e fornecedores internacionais. Muitas vezes, os cientistas e demais profissionais que atuam no país são obrigados a interromper programas de monitoramento ou de pesquisa de grande importância (estratégicos para o país) devido a uma série de fatores tais como: atrasos no processo de importação, falta de peças de reposição, ausência ou precariedade no atendimento especializado das empresas, descontinuidades no processo produtivo das sondas importadas, dentre outros fatores.

O presente projeto tem como objetivo principal o desenvolvimento de uma plataforma computacional que possibilitará o controle e o monitoramento da rede subaquática de sensores Hydronode (Pinto et al., 2012; Vieira et al., 2012), criando uma plataforma brasileira de sistemas de monitoramento robusta com capacidade para competir com tecnologias similares desenvolvidas na Alemanha, Japão ou EUA.

Nesse projeto, serão desenvolvidos os programas de controle da sonda. Especificamente, iremos trabalhar a implementação de um sistema de controle para a aquisição de dados e gerenciamento do Hydronode. Serão elaborados os módulos de gerenciamento da Acquisition Board e da Manager Board.

Finalmente, serão trabalhadas em módulos de agregação e transformação das informações recolhidas pelos sensores. Essa etapa do trabalho é a mais próxima ao usuário final da sonda. Nessa etapa iremos criar os mecanismos para comunicação em tempo real com a estação de sensoriamento e, em seguida, mecanismos para o armazenamento das informações em grandes repositórios de dados.

3 Trabalhos Relacionados

Atualmente existem *softwares* de controle e gerência de RSA comerciais, como exemplo temos o Ecowatch (YSI, 2014), *software* distribuído junto com o nó sensor 6-Series da empresa YSI, destinado para usuários da plataforma MS Windows. Por meio do Ecowatch o usuário pode realizar toda configuração e leitura dos nós sensores. Toda comunicação entre computador e nó sensor é realizada através de um cabo de comunicação serial. A Figura 3.1 apresenta a tela do sistema após configurado o nó sensor com a porta serial.

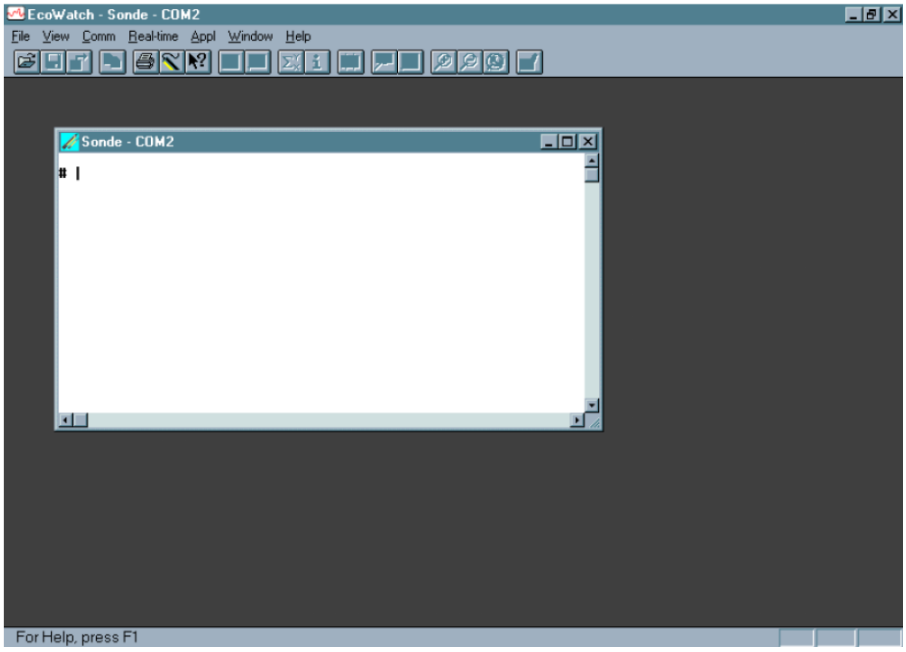

Figura 3.1: Tela inicial do sistema EcoWatch. (YSI, 2014)

Toda interface do Ecowatch é dada através de um console, em que o usuário inicialmente entra com a opção desejada e segue a utilização do sistema. A Figura 3.2 ilustra uma operação básica de configuração do nó 6-Series do Ecowatch. Essa arquitetura de interface é menos propensa a erros do que um sistema que apresenta a interação através de linhas de comando. Ainda assim essa arquitetura de interface apresenta dificultades nos quesitos de usabilidade, usuários podem ser afogados em informação, podendo apresentar dificuldades de realizar tarefas espécificas no sistema; além de ser tedioso todo o processo de realização das tarefas.

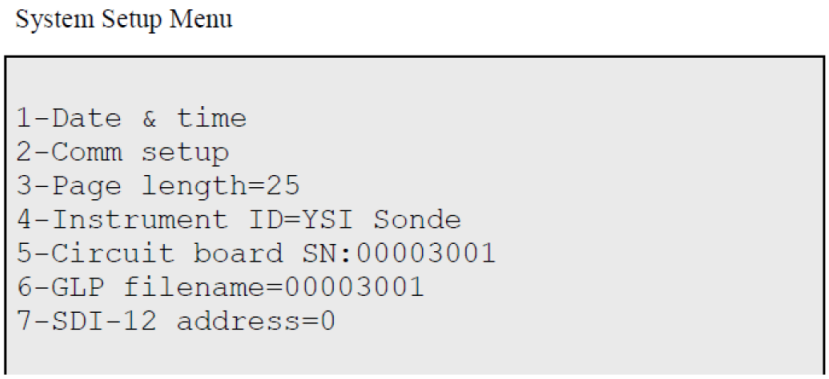

Figura 3.2: Menu principal do sistema EcoWatch. (YSI, 2014)

Desenvolvido em paralelo com este trabalho está o Hydromobile. Utilizando o nó Hydronode (Lucas da Silva e Angelica Lima, 2014), o Hydromobile é utilizado para fazer toda comunicação com o nó Hydronode atráves da transmissão *bluetooth*. O aplicativo para gerenciamento de RSA desenvolvido tem o objetivo de fornecer diversas funções para o manuseio da sonda aquática, a fim de facilitar a configuração das tarefas da rede e a coleta de dados. A Figura 3.3 apresenta a tela de leitura dos dados dos sensores na aplicação Hydromobile.

Figura 3.3: Tela de leitura dos dados dos sensores do aplicativo Hydromobile. (Lucas da Silva e Angelica Lima, 2014)

Apesar de possuir uma interface agradável, esse aplicativo foi desenvolvido visando a comunicação bluethooh e disponível apenas para o sistema operacional Android, apresentando problemas na portabilidade. Além disso, nem todas as configurações e funções de leitura do hydronode podem ser realizadas através do Hydromobile.

4 Hydronode

Nesta seção está descrito o hardware do HydroNode. A Figura 4.1 apresenta uma modelagem interna de um nó HydroNode. De acordo com a Figura 4.1, o projeto do nó foi subdividido essencialmente em 4 módulos, descritos a seguir:

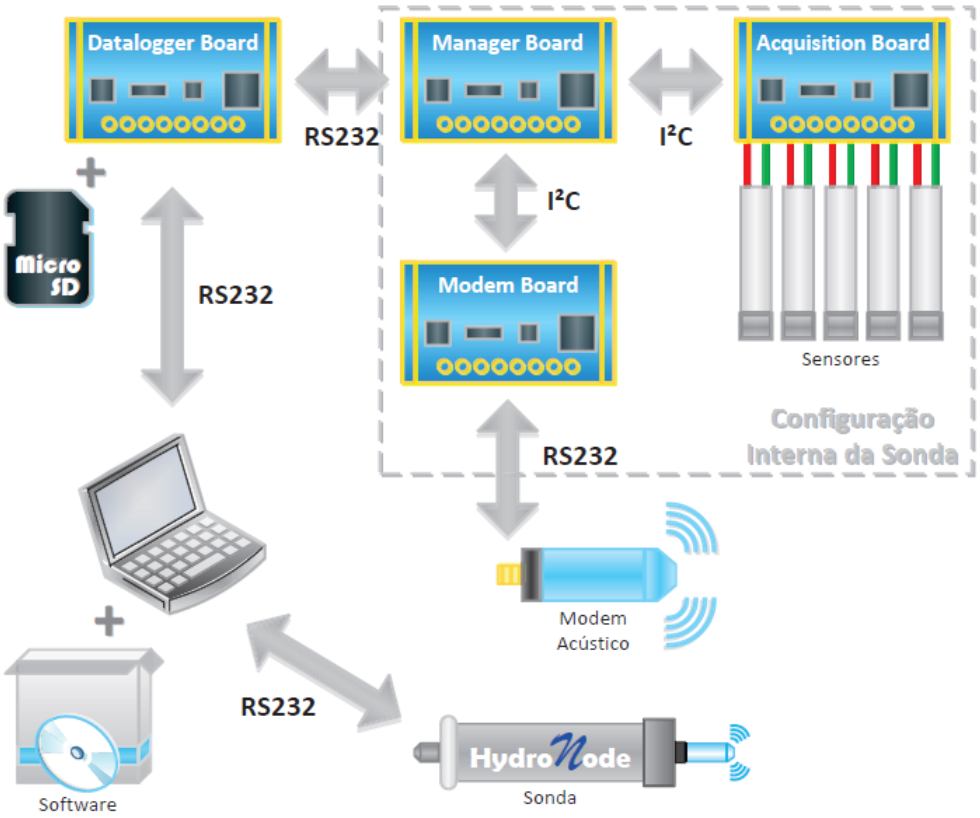

Figura 4.1: Diagrama de blocos do projeto do nó HydroNode. (Vieira et al., 2009)

Acquisition Board: Corresponde ao módulo coletor de dados. Nessa placa estão acoplados todos os sensores de monitoramento da qualidade da água;

Manager Board: Corresponde ao módulo processador de dados. Além disso, essa placa provê a gestão de funcionamento do nó, reunindo e conformando todas as informações oriundas dos outros módulos e coordenando as ações dos mesmos. Através dessa placa que é realizado o envio de constantes de calibração dos sensores e da bateria, configuração de RTC (Real-time Clock), controle da EEPROM e configuração dos

parâmetros do modem e da rede. Podemos considerar esta a placa mais importante do nó, todos os dados que chegam ou têm que sair da sonda passam por ela.

Modem Board: Corresponde ao módulo de comunicação sem fio e serial do nó. É nessa placa que se encontra acoplado o modem acústico, responsável pela troca de informação entre os elementos da rede;

Datalogger Board: Esse módulo é externo ao nó. A placa fica atrelada ao chamado nó principal da rede, o qual desempenha o papel de ponto de acesso, e armazena todas as informações próprias e dos outros nós no datalogger;

A Figura 4.2 ilustra a arquitetura de uma RSA composta por nós Hydronode. O HydroNode pode ser configurado como sensor, roteador ou porta de entrada. Agindo como um sensor, o HydroNode coleta dados de sensoriamento do ambiente. Coletados os dados, eles são armazenados e transmitidos por um módem acústico. Os nós roteadores recebem os dados transmitidos, encaminhando os pacotes para outros roteadores ou para a porta de entrada. Quando os pacotes de dados chegam na porta de entrada os dados são armazenados em um registrador presente no Datalogger ou enviados através de uma comunicação serial, *wireless* ou Wi-fi. Neste trabalho, os dados armazenados estarão disponíveis em um computador pessoal por meio de uma interface de comunicação serial.

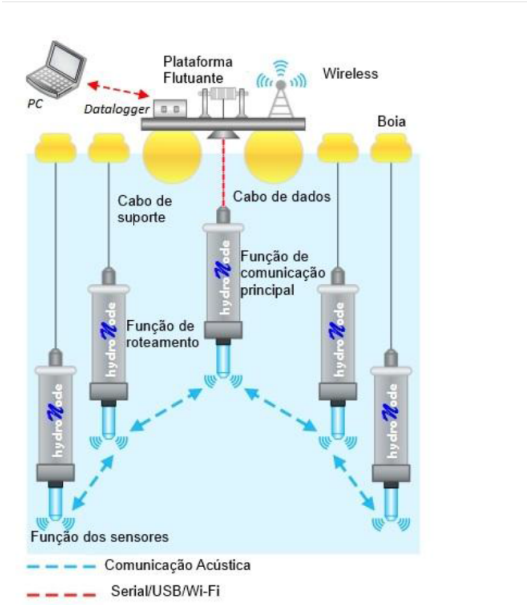

Figura 4.2: Diagrama Esquemático - Configuração da rede HydroNode (Viana et al., 2012)

5 Materiais e Métodos

5.1 Modelo de Processo

Um processo de *software* é um conjunto de atividades que levam à produção de um produto de *software*. Neste trabalho optou-se por um desenvolvimento evolucionário. Às vezes taxado de abordagem incremental, o modelo evolutivo descreve um processo na qual o *software* deve ser desenvolvido de forma a evoluir a partir de protótipos iniciais. Um esboço do sistema é desenvolvido rapidamente de acordo com as necessidades do cliente e vai sendo testado, validado e refinado com o uso. A ação inicial trata de expor o resultado ao comentário do usuário e fazer seu aprimoramento por meio de muitas versões, até que um sistema adequado tenha sido desenvolvido. Esse modelo está intimamente ligado com a ideia de prototipação.

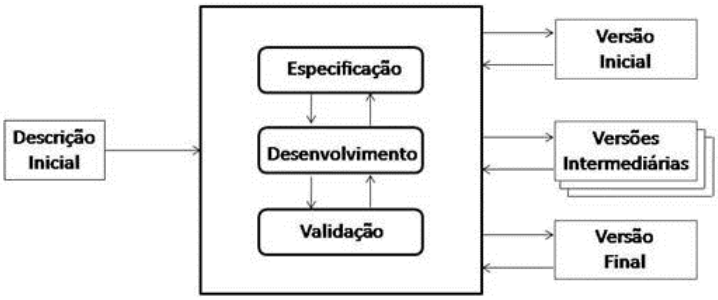

Figura 5.1: Arquitetura geral do Modelo Evolucionário

A prototipação consiste na construção experimental de um sistema de maneira rápida e com baixos custos de forma que o usuário possa avaliá-lo. O protótipo é uma versão funcional do sistema ou de parte dele (Fernandes et al., 2011).

Uma vez disponibilizado, o protótipo vai sendo refinado até chegar numa versão final que atenda completamente às necessidades do usuário(Laudon et al., 2011). Nessa abordagem temos grande vantagem de permitir a verificação antecipada do produto final por clientes e usuários, permitindo a correção dos problemas detectados.

Com o intuito de evitar problemas em futuras manutenções do *software* foi desenvolvido, em paralelo a este trabalho, um *framework* que disponibiliza de maneira fácil e intuitiva todas as mensagens presentes no protocolo de comunicação descritos no Apêndice A, bem como todo suporte no desenvolvimento de novas mensagens.

5.2 Comunicação Serial

O HydroNode pode usar apenas modem com interface serial, visto esta restrição de hardware, neste trabalho foi utilizado a biblioteca RXTX que disponibiliza para o Java Development Toolkit (JDK) uma forma de se realizar comunicação serial e paralela através das portas COM.

5.3 Bibliotecas gráficas

Atualmente, o Java suporta, oficialmente, dois tipos de bibliotecas gráficas: AWT e Swing. Neste trabalho optamos por utilizar a Swing. Inclusas em qualquer JRE ou JDK, a utilização dessa biblioteca favorece, ao máximo, o lema de portabilidade da plataforma Java. Todas as classes e métodos do Swing foram desenvolvidos tendo em mente o máximo de portabilidade possível. Com Swing, não importa qual sistema operacional, qual resolução de tela, ou qual profundidade de cores: nosso sofware se comportará da mesma forma em todos os ambientes.

A biblioteca Chart2D foi utilizada em conjunto com a Swing. Essa biblioteca fornece métodos e classes para a criação de gráficos bidimensionais que são tratados como componentes swing; assim essa biblioteca permitiu a criação de gráficos em tempo real que facilitam a visualização dos dados no *software*.

5.4 Protocolo de Comunicação

Tratando-se de comunicação entre as entidades, o protocolo desempenha um importante papel na definição de como se dará a comunicação entre as partes envolvidas.

Um protocolo é composto de um conjunto de regras que devem ser seguidas por todas as entidades envolvidas de modo a estabelecer um padrão. Esse conjunto de regras estabelece como uma entidade conecta-se a outra, como se identifica, quando pode enviar ou receber informações e quanto tempo pode esperar para que cada evento ocorra, bem como a forma de se desfazer a conexão. Uma vez que todas as entidades estão configuradas com os mesmos parâmetros e obedecem aos mesmos padrões pré-estabelecidos conseguimos assegurar que a comunicação pode ser realizada sem erros.

Como indicado por Viana et al. (2012), a comunicação será realizada através de uma interface serial. Através de um vetor de bytes, a Manager board presente na sonda hydronode interpretará a mensagem, despertando do estado de hibernação, se nescessário, a Aquisition Board.

Assim como qualquer protocolo, toda mensagem enviada contém um padrão pré-estabelecido. O Apêndice A deste trabalho apresenta, em detalhes, toda a especificação do protocolo, incluindo as mensagens que podem ser trocadas entre as entidades.

Abaixo é apresentado, como exemplo, a principal mensagem utilizada para a comunicação entre a sonda e o *software*. Todas as mensagens utilizadas seguem um padrão semelhante a esse.

5.4.1 Selecionar dados em tempo real

Mensagem de dados enviada pelo Software:

É enviado um byte com o comprimento do resto da mensagem ($LENGTH$); um byte com o número 1, um byte com a identificação da sonda (ID_NODE). A sonda deve enviar dados em tempo real. A estrutura da mensagem a ser enviada é:

BYTE 0	BYTE 1	BYTE 2
LENGTH	1	ID_NODE

Mensagem de dados enviada pela sonda como resposta:

Essa é a mensagem enviada pela sonda contendo os dados dos sensores. Será enviada no caso de aquisição em tempo real. Ela contém o ID do nó a que os dados se referem (ID_NODE); o valor de segundos desde 1 de janeiro de 1970, em quatro

bytes ($DATE\_Bx$); um byte que informa quais os sensores cujos dados serão enviados (S_ENB); os dados dos sensores, em 2 bytes (Sx_Bx); o status de carga do nó, de 1 a 100% (SOC).

A estrutura da mensagem a ser enviada é:

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
ID_NODE	DATE_B0	DATE_B1	DATE_B2	DATE_B3	S_ENB	S1_B0	S1_B1

BYTE 8	BYTE 9	BYTE 10	BYTE 11	BYTE 12	BYTE 13	BYTE 14
S2_B0	S2_B1	S3_B0	S3_B1	S4_B0	S4_B1	S5_B0

BYTE 15	BYTE 16	BYTE 17	BYTE 18	BYTE 19	BYTE 20	BYTE 21
S5_B1	S6_B0	S6_B1	S7_B0	S7_B1	S8_B0	S8_B1

BYTE 22
SOC

Mensagem de confirmação enviada pela sonda:

Depois de cada mensagem, a sonda confirma sua recepção completa com um byte acknowledge (ACK). Esse byte indica o sucesso do envio, bem como o atraso para o *software* até que ele possa enviar outra mensagem. O atraso é calculado multiplicando o número inteiro que representa o byte de 100ms. Se nenhum ACK for recebido dentro de 10s a mensagem será enviada novamente. Se após a 12ª vez náo for recebido um ACK, uma mensagem de erro é exibida.

A estrutura da mensagem a ser enviada é:

BYTE 0
ACK

Depois de cada pacote de dados, é recebido pelo aplicativo uma outra mensagem de 2 bytes. Deve-se enviar um byte com o comprimento do resto da mensagem ($LENGTH$) e um byte ($MAIS$) indica se devem ser enviadas mais leituras com o valor 1 ou 0 para parar de enviar.

A estrutura da mensagem a ser enviada é:

BYTE 0	BYTE 1
LENGTH	MAIS

A mensagem de resposta enviada pela sonda conterá dois bytes: Um de quantidade de bytes ($LENGTH$) e um de configuração realizada com sucesso.

BYTE 0	BYTE 1
LENGTH	ACK

5.5 Arquitetura do Software

Antes de realçar as funcionalidades do *software* é importante explicar como funciona a arquitetura do *software*. Por uma maneira de facilitar a manutenção do código, como já dito anteriormente, procuramos desenvolver o *software* em camadas. O projeto está dividido em três principais camadas: a camada de comunicação, a camada de controle e finalmente a camada de interface.

Camada de Comunicação: Nessa camada foi utilizada a biblioteca RXTX. O objetivo dessa camada é que ocorra toda a manipulação das portas seriais, tratando do empacotamento e da recepção das mensagens. Nessa camada conseguimos abrir portas seriais, ler e enviar dados através de uma determinada porta.

Camada de Controle: É nessa camada que está implementado todo o protocolo presente no Anexo A. O corpo das mensagens é enviado para a camada de comunicação, no qual é realizado o envio das mesmas. São esperadas ter as maiores modificações futuras no sistema presentes nessa camada, então o *framework* desenvolvido se aplica aqui, visando assim facilitar a manutenção do *software*.

Camada de Interface: Como descrito no próprio nome, é nessa camada que está a interface do sistema. É aqui que está presente toda a codificação da interação usuário-computador. Presentes nessa camada também, todas as telas do sistema, desde a apresentação de dados até as telas de configuração.

O objetivo dessa arquitetura é minimizar a dependência entre as camadas de *software*. Dessa forma, propomos um sistema com grande manutenibilidade e testabilidade, fatores essenciais para qualidade de *software*.

6 Resultados

As figuras abaixo apresentam as principais telas do aplicativo desenvolvido.

Figura 6.1: Tela Completa do Sistema

A Figura 6.2 apresenta a tela de configuração de tipo dos sensores. Essa figura apresenta uma interface simples, utilizando apenas as classes da biblioteca swing. Esta tela foi desenhada para o usuário informar qual o nó que deseja configurar, qual o sensor e também informa o novo tipo. Após o envio da mensagem, uma confirmação surge informando o sucesso ou falha da operação.

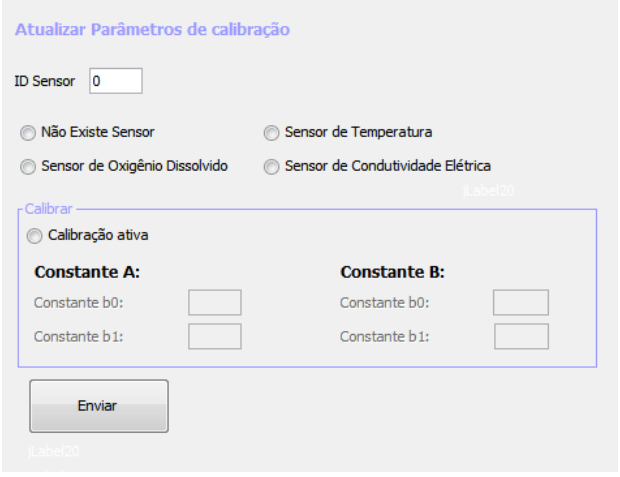

Figura 6.2: Tela de configuração de tipo dos sensores

A figura 6.3 apresenta a tela de habilitação ou desabilitação dos sensores cadastrados. Informado o id do nó, o sistema apresenta o estado dos sensores do nó selecionado, permitindo ao usuário modificar esses estados se necessário.

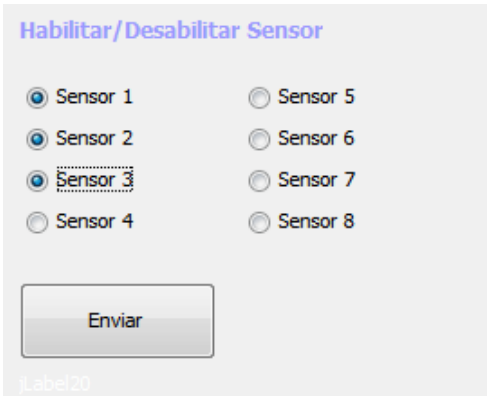

Figura 6.3: Tela de habilitação/desabilitação dos sensores

A Figura 6.4 é uma das telas de maior importância em nosso sistema, pois a partir dela que o usuário inicia uma leitura dos dados coletados pelos sensores em tempo real. Na Área de Texto presente no centro dessa tela é gerado o log em tempo real da leitura.

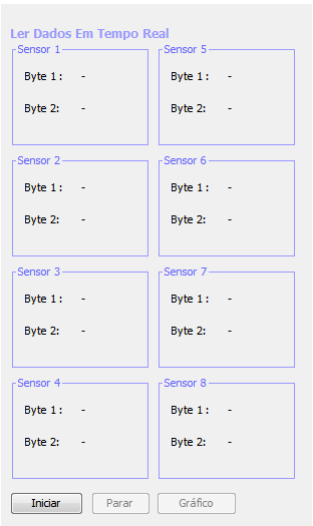

Figura 6.4: Tela de leitura dos dados dos sensores

Para uma melhor visualização dos dados, basta o usuário selecionar a opção gráficos, no qual utilizando as classes presentes na biblioteca Chart2D, construimos a tela presente na Figura 6.5. No centro da tela, são gerados gráficos em tempo real, pelos quais o usuário tem a opção de escolher quais sensores deseja acompanhar na leitura.

Figura 6.5: Tela de gráfico da leitura dos dados dos sensores em tempo real.

A Figura 6.6 apresenta a tela de carga de bateria. Nessa tela pode-se verificar qual a porcentagem restante de bateria de uma sonda selecionada.

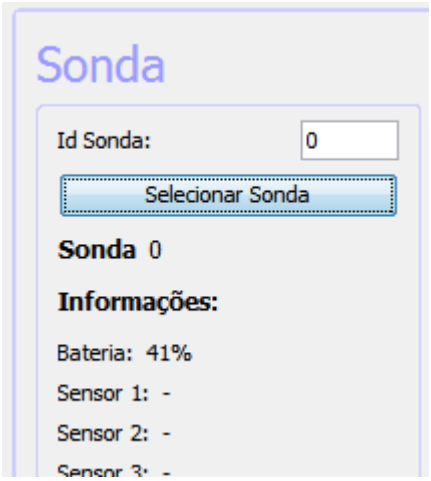

Figura 6.6: Tela de carga de bateria

O *software* pode ser encontrado em sites.google.com/site/softwarecontrolebruno/.

7 Conclusão e Trabalhos Futuros

Neste trabalho foi desenvolvido uma plataforma de controle para o Hydronode, uma rede de nós sensores aquáticos de múltiplos propósitos. Descreveu-se as funcionalidades básicas de hardware, como a Aquisition Board e a Manager Board, bem como o funcionamento básico dessa rede. Determinados os princípios básicos de hardware, todas as bibliotecas utilizadas pelo *software* foram explicitadas, bem como todo processo de produção do *software*, em que seguindo as boas maneiras da engenharia de *software*, buscou-se minimizar o risco de sua produção. Por fim, foi apresentado a interface do *software* desenvolvido, explicitando as principais telas do sistema.

O presente trabalho trata de um problema em aberto ainda no Brasil. Há falta de uma tecnologia nacional para o monitoramento de regiões aquáticas e o interesse crescente que a área vem adquirindo nos últimos anos, criou-se uma grande lacuna de depêndencia tecnológica das pesquisas brasileiras em plataformas estrangeiras de RSA. Este trabalho busca minimizar a lacuna que o Brasil possui na área de RSA, desenvolvendo uma plataforma brasileira para o monitoramento das regiões aquáticas do país. Como trabalhos futuros, visamos a manutenção do *software*, onde desenvolvemos um framework para auxiliar para este trabalho, devido a necessidade de que novas funções possam ser requisitadas a qualquer momento. Procuramos também trabalhar em melhorias na interface do *software* de maneira a ficar mais agradável para o usuário.

Referências Bibliográficas

Akyildiz, I. F.; Pompili, D. ; Melodia, T. Underwater acoustic sensor networks: research challenges. **Ad hoc networks**, v.3, n.3, p. 257–279, 2005.

Akyildiz, I. F.; Melodia, T. ; Chowdury, K. R. **Wireless multimedia sensor networks: A survey**, 2007.

Chandrasekhar, V.; Seah, W. K.; Choo, Y. S. ; Ee, H. V. **Localization in underwater sensor networks: survey and challenges**. In: Proceedings of the 1st ACM international workshop on Underwater networks, p. 33–40. ACM, 2006.

Fernandes, A. A.; TEIXEIRA, D. d. S. Fábrica de software. **São Paulo: Atlas**, 2004.

Heidemann, J.; Stojanovic, M. ; Zorzi, M. Underwater sensor networks: applications, advances and challenges. **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences**, v.370, n.1958, p. 158–175, 2012.

Kong, J.; Cui, J.-h.; Wu, D. ; Gerla, M. **Building underwater ad-hoc networks and sensor networks for large scale real-time aquatic applications**. In: Military Communications Conference, 2005. MILCOM 2005. IEEE, p. 1535–1541. IEEE, 2005.

Laudon, K. C.; Laudon, J. P. ; others. **Essentials of management information systems**. Pearson Upper Saddle River, 2011.

Loureiro, A. A.; Nogueira, J. M. S.; Ruiz, L. B.; de Freitas Mini, R. A.; Nakamura, E. F. ; Figueiredo, C. M. S. **Redes de sensores sem fio**. In: Simpósio Brasileiro de Redes de Computadores (SBRC), p. 179–226, 2003.

da Silva e Angelica Lima, L. **Hydromobile: Comunicaçã com nós sensores aquáticos utilizando dispositivos móveis**. 2014.

Nayak, A.; c, I. S. **Wireless sensor and actuator networks**. Wiley Online Library, 2010.

Pinto, D.; Viana, S. S.; Nacif, M.; Augusto, J.; Vieira, L. F.; Vieira, M. A.; Vieira, A. B. ; Fernandes, A. O. **Hydronode: A low cost, energy efficient, multi purpose node for underwater sensor networks**. In: Local Computer Networks (LCN), 2012 IEEE 37th Conference on, p. 148–151. IEEE, 2012.

Viana, S. S.; Silva, L. B.; Lima, A. S.; VieiraΨ, A. B.; Vieira, L. F.; Vieira, M. A.; Fernandes, A. O. ; Nacif, J. A. M. Hydronode: Uma rede de sensores aquáticos de baixo custo e consumo.

VIEIRA, VIEIRA, N. V. **Desenvolvimento e teste de uma plataforma computacional aquática para mensuração contínua de variáveis limnológicas de qualidade de água de reservatórios tropicais**. Universidade Federal de Minas Gerais.

Vieira, L. F. M. **Underwater sea swarm**. University of California at Los Angeles, 2009.

Vieira, L. F.; Vieira, M. A.; Pinto, D.; Nacif, J. A. M.; Viana, S. S. ; Vieira, A. B. **Hydronode: an underwater sensor node prototype for monitoring hydroelectric reservoirs**. In: Proceedings of the Seventh ACM International Conference on Underwater Networks and Systems, p. 43. ACM, 2012.

YSI. Disponivel em: http://www.ysi.com/productsdetail.php?EcoWatch-Software-15.

Apêndice A

Este documento contém a descrição das mensagens que serão trocadas pelo PC e pela sonda, tendo como intermediária a Datalogger Board. As mensagens que constituem o protocolo de comunicação estão separadas em tópicos. Por exemplo: As solicitações do PC para a sonda estão separadas das respostas às mesmas, da sonda para o PC.

1. Comunicação da sonda via serial: mensagens enviadas pelo PC

Neste tópico estão descritas as solicitações que o PC envia para a Sonda. Todas as mensagens terão como resposta uma mensagem de confirmação, seguida pela resposta específica quando houver, da forma descrita.

1.1. Selecionar dados em tempo real:

Deve enviar um byte com o comprimento do resto da mensagem (LENGTH); um byte com o número 1, um byte com a identificação da sonda (ID_NODE). A sonda vai enviar dados em tempo real.

A estrutura da mensagem a ser enviada é:

BYTE 0	BYTE 1	BYTE 2
LENGTH	1	ID_NODE

A sonda responderá com uma mensagem de ack e em seguida enviará as leituras em um pacote de dados (vide tópico 2).

Depois de cada pacote de dados recebido, deve ser enviada uma outra mensagem de 2 bytes, contendo um byte com o comprimento do resto da mensagem (LENGTH) e um byte (MAIS) indicando se devem ser enviadas mais leituras (valor 1) ou se o envio deve ser encerrado (valor 0).

A estrutura da mensagem a ser enviada é:

BYTE 0	BYTE 1
LENGTH	MAIS

1.2. Configuração dos sensores:

Deve enviar um byte com o comprimento do resto da mensagem (LENGTH); um byte com o número 2; um byte com a identificação da sonda (ID_NODE); um byte de opção (OPTION). As opções são:

OPTION = 1: habilitar desabilitar sensores
OPTION = 2: calibrar sensor;
OPTION = 3: configurar tipo do sensor
OPTION = 4: ler tipo do sensor.

1.2.1. Habilitar ou desabilitar sensores:

Para habilitar ou desabilitar um sensor, deve ser enviado após o byte de opção (OPTION) um byte (HAB) cujos bits informam se os sensores estão ativados (1) ou desativados (0). O bit zero dá informações sobre o sensor 1, o bit um, sobre o sensor 2; e assim por diante.

A estrutura da mensagem a ser enviada é:

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4
LENGTH	2	ID_NODE	OPTION	HAB

1.2.2. Calibrar Sensor

Para atualizar os parâmetros de calibração dos sensores deve ser enviado após o byte de opção (OPTION) um byte com o ID do sensor (ID_SENSOR) seguido de mais quatro bytes com as constantes conforme a estrutura de mensagem abaixo:

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7	BYTE 8
LENGTH	2	ID_NODE	OPTION	ID_SENSOR	CONST_A_B0	CONST_A_B1	CONST_B_B0	CONST_B_B1

1.2.3. Configurar tipo do sensor:

Para configurar o tipo do sensor, devem ser enviados após o byte de opção (OPTION), dois bytes: o primeiro byte informa qual sensor deve ser configurado seguindo a mesma ideia da mensagem de habilitação, ou seja, cada bit do byte é um sensor. Nessa mensagem só pode ser configurado um sensor de cada vez. O segundo byte é o tipo do sensor seguindo as definições abaixo:

TYPE = 0: Não existe sensor nessa porta. Configuração default;
TYPE = 1: O sensor ligado a esta porta é um sensor de Temperatura;
TYPE = 2: O sensor ligado a esta porta é um sensor de Condutividade elétrica;
TYPE = 3: O sensor ligado a esta porta é um sensor de Oxigênio dissolvido;
Obs: Podem ser adicionados novos tipos de sensores.
A estrutura da mensagem a ser enviada é:

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5
LENGTH	2	ID_NODE	OPTION	ID_SENSOR	TYPE

1.2.4. Ler o tipo do sensor:

Para ler o tipo do sensor deve ser enviado após o byte de opção (OPTION) um byte que informa qual o sensor cujo tipo deve ser lido, seguindo a mesma ideia da mensagem de habilitação, ou seja, cada bit do byte é um sensor, nessa mensagem só pode ser lido o tipo de um sensor de cada vez.

A estrutura da mensagem a ser enviada é:

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4
LENGTH	2	ID_NODE	OPTION	ID_SENSOR

1.3. Configurar ID da sonda:

Deve enviar um byte com o comprimento do resto da mensagem (LENGTH); um byte com o número 4; um byte com a identificação da sonda cujo ID se deseja alterar (ID_NODE) e um byte com a nova identificação da sonda (NOVO_ID).

A estrutura da mensagem a ser enviada é:

BYTE 0	BYTE 1	BYTE 2	BYTE 3
LENGTH	4	ID_NODE	NOVO_ID

1.4. Configurar data e hora:

Deve ser enviado um byte com o comprimento do resto da mensagem (LENGTH); um byte com o número 5; um byte com a identificação da sonda (ID_NODE) e a data, em quatro bytes, informada em segundos contados desde 1 de janeiro de 1970, às zero horas.

A estrutura da mensagem a ser enviada é:

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6
LENGTH	5	ID_NODE	DATE_B0	DATE_B1	DATE_B2	DATE_B3

1.5. Definir intervalo de tempo de aquisição:

Deve enviar um byte com o comprimento do resto da mensagem (LENGTH); um byte com o número 6; um byte com a identificação da sonda (ID_NODE) e o intervalo de aquisição, em dois bytes. A estrutura da mensagem a ser enviada é:

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4
LENGTH	6	ID_NODE	TIME_INTERVAL_B0	TIME_INTERVAL_B1

1.6. Verificar status da memória EEPROM, ou formatá-la:

Deve enviar um byte com o comprimento do resto da mensagem (LENGTH); um byte com o número 7; um byte com a identificação da sonda (ID_NODE); e um byte informando a opção a ser realizada.

Os seguintes números devem ser enviados, de acordo com a opção:

OPTION = 1: ler memória;

OPTION = 2: verificar status;

OPTION = 3: formatar.

A estrutura da mensagem a ser enviada é:

BYTE 0	BYTE 1	BYTE 2	BYTE 3
LENGTH	7	ID_NODE	OPTION

Obs: No caso da opção 1, as leituras serão enviadas segundo a descrição do tópico 2.4.

1.7. Configurar modem acústico:

Deve enviar um byte com o comprimento do resto da mensagem (LENGTH); um byte com o número 8; um byte com a identificação da sonda (ID_NODE); um byte (OPTION) que informa qual operação deseja realizar; bytes com dados. São elas:
OPTION = 1: Atualizar RTT em segundos;
OPTION = 2: Atualizar Taxa de transmissão da dados (quantas vezes por hora);
OPTION = 3: Configurar intervalos de transmissão de dados;
OPTION = 4: Configurar intervalos de recepção de dados;
OPTION = 5: Configurar os timeouts do protocolo de comunicação acústica;
OPTION = 6: Configurar o número máximo de retransmissões;
OPTION = 7: Setar modo no modem;
OPTION = 8: Setar velocidade de transmissão no modem;
OPTION = 9: Setar velocidade de recebimento do modem;
OPTION = 10: Setar threshold do modem;
As opções são descritas em maior detalhe nos tópicos seguintes.

A estrutura da mensagem a ser enviada é:

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	...	BYTE X
LENGHT	8	ID_NODE	OPTION	DADOS	...	DADOS

1.7.1. Atualizar RTT em segundos

Envia o tempo de RTT para cada nó, com tamanho de dois bytes. A mensagem completa deverá ser:

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5
LENGTH	8	ID_NODE	1	RTT_B0	RTT_B1

1.7.2. Atualizar Taxa de transmissão de dados (quantas vezes por hora)

Envia a taxa de transmissão de dados para cada nó. A mensagem completa deverá ser:

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4
LENGTH	8	ID_NODE	2	RATE

1.7.3. Configurar intervalos de transmissão de dados

Envia o intervalo de transmissão de dados, em segundos, para cada nó, com tamanho de dois bytes. A mensagem completa deverá ser:

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5
LENGTH	8	ID_NODE	3	TRANS_B0	TRANS_B1

1.7.4. Configurar intervalos de recepção de dados

Envia o intervalo de recepção de dados, em segundos, para cada nó, com tamanho de dois bytes.

A mensagem completa deverá ser:

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5
LENGTH	8	ID_NODE	4	RECV_B0	RECV_B1

1.7.5. Configurar os timeouts do protocolo de comunicação acústica

Envia o tempo de timeout para cada nó, com tamanho de dois bytes. A mensagem completa deverá ser:

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5
LENGTH	8	ID_NODE	5	TIMEOUT_B0	TIMEOUT_B1

1.7.6. Configurar o número máximo de retransmissões

Envia o número máximo de retransmissões para cada nó, com tamanho de um bytes. A mensagem completa deverá ser:

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 6
LENGTH	8	ID_NODE	6	NUM_RETRANSMIT

1.7.7. Setar modo no modem

Seta cada modem em modo de controle para cada nó, com tamanho de um byte cujos bits informam se o modo estão modo de controle (0) ou modo de dados (1). A mensagem completa deverá ser:

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4
LENGTH	8	ID_NODE	7	MODO

1.7.8. Setar velocidade de transmissão no modem

Envia a velocidade de transmissão de cada modem. A mensagem completa deverá ser:

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4
LENGTH	8	ID_NODE	8	TRANSMIT_VEL

1.7.9. Setar velocidade de recebimento do modem

Envia a velocidade de recebimento de cada modem. A mensagem completa deverá ser:

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4
LENGTH	8	ID_NODE	9	RECV_VEL

1.7.10. Setar trheshold do modem

Seta o threshold de cada modem. A mensagem completa deverá ser:

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4
LENGTH	8	ID_NODE	10	THRESHOLD

1.8. Configuração da bateria:

Deve enviar um byte com o comprimento do resto da mensagem (LENGTH); um byte com o número 9; um byte com a identificação da sonda (ID_NODE); um byte de opção (OPTION) e os dados conforme a opção.

OPTION = 1: Atualizar parâmetros de medição da carga da bateria;
OPTION = 2: Ler estado da carga da bateria;

1.8.1. Atualizar constantes da bateria:

Para atualizar as constantes da bateria deve ser escolhido a opção 1 e enviar 10 parâmetros de dois bytes para a conversão dos dados de tensão da bateria em dados de carga restante.

A estrutura da mensagem a ser enviada para a opção 1 é:

```
|  BYTE 0  |  BYTE 1  |  BYTE 2  |  BYTE 3  |  BYTE 4  |  BYTE 5  |  BYTE 6  |
-------------------------------------------------------------------------------
|  LENGTH  |    9     |  ID_NODE | OPTION=1| SOC_1_B0 | SOC_1_B1 | SOC_2_B0 |

|  BYTE 7  |  BYTE 8  |  BYTE 9  |  BYTE 10 |  BYTE 11 |  BYTE 12 |  BYTE 13 |
-------------------------------------------------------------------------------
| SOC_2_B1 | SOC_3_B0 | SOC_3_B1 | SOC_4_B0 | SOC_4_B2 | SOC_5_B0 | SOC_5_B1 |

|  BYTE 14 |  BYTE 15 |  BYTE 16 |  BYTE 17 |  BYTE 18 |  BYTE 19 |  BYTE 20 |
-------------------------------------------------------------------------------
| SOC_6_B0 | SOC_6_B1 | SOC_7_B0 | SOC_7_B1 | SOC_8_B0 | SOC_8_B1 | SOC_9_B0 |

|  BYTE 21  |  BYTE 22  | BYTE 23  |
-----------------------------------
| SOC_9_B1 | SOC_10_B0  | SOC_10_1 |
```

1.8.2. Ler estado da bateria:

Para ler o estado da bateria deve escolher a opção 2 e não precisa dos bytes de dados porque é somente para ler estado da carga da bateria.

A estrutura da mensagem a ser enviada para a opção 2 é:

```
|  BYTE 0  |  BYTE 1  |  BYTE 2  |  BYTE 3   |
-----------------------------------------------
|  LENGTH  |    9     |  ID_NODE | OPTION=2  |
```

2. Comunicação da sonda via serial – mensagens recebidas pelo PC

2.1. Mensagem de confirmação

Depois de qualquer mensagem cuja resposta não esteja descrita abaixo, a sonda confirmará a sua recepção completa com um ACK. O byte de ACK, além de confirmar a recepção, indicará também o atraso para o software PC até que ele possa enviar outra mensagem. O atraso é representado em centenas de milissegundos. Em outras palavras, para calcular o atraso em segundos basta multiplicar o valor representado pelo ACK por 100.

Se nenhum ACK for recebido dentro de 10 segundos, a mensagem será enviada novamente. Se, após a 12 ª tentativa, não for recebido um ACK, uma mensagem de erro será exibida.

A estrutura da mensagem de confirmação é:

BYTE 0	BYTE 1
LENGTH	ACK

2.2. Mensagem de dados

Esta será contém os dados dos sensores. Será enviada no caso de aquisição em tempo real, do tópico 1. Ela contém, além do tamanho do resto da mensagem, o ID do nó ao qual os dados se referem (ID_NODE); o valor em segundos, desde 1 de janeiro de 1970, em quatro bytes (DATE_Bx); um byte que informa quais os sensores h (S_ENB); os dados dos sensores, em 2 bytes (Sx_Bx); o status de carga do nó, de 1 a 100% (SOC).

No uso de (S_ENB), o seguinte exemplo deve ser considerado. Caso seu valor seja, por exemplo, 00100010, então serão enviados apenas os dados dos sensores 3 e 7, ou seja, o bit mais à esquerda refere-se ao sensor 1. O tamanho da mensagem, então, é fixo, e S_ENB indicará quais dados da mensagem deverão ser capturados.

A estrutura da mensagem a ser enviada é:

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
LENGTH	ID_NODE	DATE_B0	DATE_B1	DATE_B2	DATE_B3	S_ENB	S1_B0

BYTE 8	BYTE 9	BYTE 10	BYTE 11	BYTE 12	BYTE 13	BYTE 14
S1_B1	S2_B0	S2_B1	S3_B0	S3_B1	S4_B0	S4_B1

BYTE 15	BYTE 16	BYTE 17	BYTE 18	BYTE 19	BYTE 20	BYTE 21
S5_B0	S5_B1	S6_B0	S6_B1	S7_B0	S7_B1	S8_B0

BYTE 22	BYTE 23
S8_B1	SOC

2.3. Mensagem de leitura do sensor

Esta mensagem é enviada em caso de solicitação de leitura do tipo do sensor (tópico 1.2.4). Conterá três bytes, um de quantidade de bytes (LENGTH) e um de confirmação e o terceiro byte informando o tipo do sensor.

BYTE 0	BYTE 1	BYTE 2
LENGTH	ACK	TYPE

2.4. Mensagem da EEPROM

2.4.1. Leitura da memória

É enviada no caso da opção 1, da mensagem de status da EEPROM (tópico 1.6). Primeiro será enviada a quantidade de leituras, em dois bytes, sendo o primeiro o mais significativo e o segundo o menos significativo, conforme abaixo:

BYTE 0	BYTE 1	BYTE 2
LENGTH	QTD_B1	QTD_B0

Após serem enviados esses bytes, a sonda aguarda uma mensagem para cada leitura a ser enviada. Esta mensagem pode ser usada para que ela pare de enviar as leituras ou continue enviando.

A estrutura da mensagem a ser enviada é:

BYTE 0	BYTE 1
LENGTH	MAIS

Onde “MAIS” autoriza o envio das próximas leituras (1 = mais leituras / 0 = parar envio).

A cada solicitação de mais pacotes de leituras (incluindo a primeira vez), um pacote de leituras da EEPROM será enviado, conforme abaixo:

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
LENGTH	ID_NODE	DATE_B0	DATE_B1	DATE_B2	DATE_B3	SEN_EN	S1_B0

BYTE 8	BYTE 9	BYTE 10	BYTE 11	BYTE 12	BYTE 13	BYTE 14
S1_B1	S2_B0	S2_B1	S3_B0	S3_B1	S4_B0	S4_B1

BYTE 15	BYTE 16	BYTE 17	BYTE 18	BYTE 19	BYTE 20	BYTE 21
S5_B0	S5_B1	S6_B0	S6_B1	S7_B0	S7_B1	S8_B0

BYTE 22	BYTE 23
S8_B1	SOC

2.4.2. Status da Memória

Enviada no caso da opção 2, (tópico 1.6). A resposta contém 3 bytes, um byte com o tamanho do resto da mensagem, um byte de ACK e o seguinte representando o valor inteiro do status da memória, em percentagem.

A estrutura da resposta é:

```
| BYTE 0 | BYTE 1 | BYTE 2 |
----------------------------
| LENGTH |  ACK   | STATUS |
```

2.5. Mensagem de estado da bateria

Será enviada em caso de leitura do estado da bateria (tópico 1.8.2). É constituída de 3 bytes sendo um length, um ack e o terceiro, a quantidade de bateria em porcentagem.

```
| BYTE 0 |  BYTE 1  |  BYTE 2          |
---------------------------------------
| LENGTH |  ACK     | CARGA_DA_BATERIA%|
```

3. Comunicação da Datalogger via Serial – Mensagens enviadas pelo PC

A Datalogger funcionará como intermediária entre a sonda e o PC. Por isso, as mensagens trocadas entre ambos serão interceptadas pela Datalogger, que também trabalhará com as mesmas. Além das mensagens entre o PC e a sonda, a Datalogger trabalhará com as mensagens a seguir:

3.1. Definir profundidade inicial e final

Deve ser enviado um byte com o tamanho do resto da mensagem; um byte com o número 32; dois bytes com a profundidade inicial e dois bytes com a profundidade final, ambas em centímetros. Por padrão, as profundidades inicial e final são 0m (mínimo) e 30m (máximo), respectivamente.

A estrutura da mensagem a ser enviada é:

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5
LENGTH	32	INITIAL_B0	INITIAL_B1	FINAL_B0	FINAL_B1

3.2. Agendamento de posição

Deve ser enviado um byte com o comprimento do resto da mensagem; um byte com o número 33; dois bytes com a distância, em centímetros, que a sonda deve descer a cada etapa; dois bytes com o tempo de permanência em cada profundidade, em segundos; dois bytes; um byte com o número de ciclos que deve ser realizado e dois bytes com o tempo de espera para o próximo ciclo.

A estrutura da mensagem a ser enviada é:

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6
LENGTH	33	DIST_B0	DIST_B1	TIME_B0	TIME_B1	CYCLES

BYTE 7	BYTE 8
SLEEP_B0	SLEEP_B1

3.3. Mandar a sonda para a profundidade desejada

Esta mensagem cancela o comando de Agendamento. Deve ser enviado um byte com o tamanho do resto da mensagem; um byte com o número 34; dois bytes com a profundidade desejada, em centímetros (máximo 3000cm).

A estrutura da mensagem a ser enviada é:

BYTE 0	BYTE 1	BYTE 2	BYTE 3
LENGTH	34	DIST_B0	DIST_B1

3.4. Solicitar dados do Cartão de Memória

Deve ser enviado um byte com o comprimento do resto da mensagem; um byte com o número 35 e um byte indicando se o conteúdo do cartão deve ser apagado ou não, onde “0” significa que o conteúdo deve ser mantido e “1”, que o conteúdo deve ser apagado.

A estrutura da mensagem a ser enviada é:

BYTE 0	BYTE 1	BYTE 2
LENGTH	35	DELETE

3.5. Configurar nome do arquivo

Deve ser enviado um byte com o tamanho do resto da mensagem; um byte com o número 36 e, em seguida, um byte por letra do nome que se deseja colocar. O tamanho da mensagem pode ser variável.

A estrutura da mensagem a ser enviada é (exemplo: configurar o nome para “dado”):

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5
LENGTH	36	'd'	'a'	'd'	'o'

3.6. Status Datalogger

Esta mensagem é usada para saber qual o estado em que se encontra a Datalogger. Ou seja, qual tarefa que ela está executando.

Deve ser enviado um byte com o tamanho do resto da mensagem; um byte com o número 37 e um byte com a opção desejada (0 = status básico, 1 = status completo). A estrutura da mensagem a ser enviada é:

BYTE 0	BYTE 1	BYTE 2
LENGTH	37	OPTION

4. Comunicação da Datalogger via serial – Mensagens recebidas pelo PC.

4.1. Mensagem de confirmação

Após cada solicitação recebida pelo PC, a Datalogger retorna um byte de ACK. No caso de mensagens do PC direcionadas para a sonda, é apenas encaminhado o byte de ACK recebido da mesma (pode ser visto na seção 2.1). No caso de mensagens específicas para Datalogger, é retornado um ACK obedecendo ao mesmo padrão da seção 2.1, representando o tempo de espera para resposta.
A estrutura da mensagem enviada se repete:

BYTE 0	BYTE 1
LENGTH	ACK

4.2. Mensagem do cartão de memória

A ser definido pelo Jean.

4.3. Mensagem do cartão de memória

Enviada em caso de solicitação de status (mensagem 37, tópico 3.6). Essa mensagem conterá uma máscara de configuração, que é uma variável de 8 bits e está descrita abaixo:

Máscara de configuração:
Bit 0: Modo agendamento (setado pelo comando 33): 0 = desligado, 1 = ligado
Bit 1: Refere-se à definição do tempo de aquisição(6): 0 = não definido, 1 = definido
Bit 2: Refere-se a configuração de data (5): 0 = nao configurado, 1 = configurado
Bit 3: Refere-se à solicitacao de modo fixo: 0 = desligado, 1 = ligado
Bit 4: Estado de espera entre um ciclo e outro: 0 = desligado, 1 = ligado
Os bits restantes ainda não estão sendo utilizados
No caso da opção 0, a mensagem conterá apenas o tamanho e a máscara acima:

BYTE 0	BYTE 1
LENGTH	CONFIG

No caso da opção 1, será enviada uma mensagem constituída por: tamanho do resto da mensagem (byte 0); máscara de configuração (byte 1); número de ciclos restantes (byte 2), profundidade inicial, mínima (byte 3 e byte 4); profundidade final, máxima (byte 5 e byte 6); profundidade atual (byte 7 e byte 8); distância descida a cada passo (bytes 9 e 10); tempo em cada profundidade (bytes 11 e 12); tempo de espera entre ciclos (bytes 13 e 14); intervalo de tempo de aquisição (byte 15 e byte 16); e mode (byte 17) que informará se os tempos passados estão em segundos[1], minutos[2] ou horas[3].

A estrutura da mensagem de resposta, então, será:

BYTE 0	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6
LENGTH	CONFIG	CYCLES	INITIAL_B0	INITIAL_B1	FINAL_B0	FINAL_B1

BYTE 7	BYTE 8	BYTE 9	BYTE 10	BYTE 11	BYTE 12	BYTE 13
CURRENT_B0	CURRENT_B1	DIST_B0	DIST_B1	TIME_B0	TIMES_B1	SLEEP_B0

BYTE 14	BYTE 15	BYTE 16	BYTE 17
SLEEP_B1	TIME_AQB0	TIME_AQB0	MODE

5. Comunicação da Datalogger via Serial – Mensagens recebidas pela Sonda

A Datalogger utilizará as mesmas mensagens entre o PC e a Sonda. Além delas, estão as mensagens a seguir:

5.1. Controle de Medição automática

Após definido o intervalo de tempo de aquisição, a Datalogger utilizará esta mensagem para iniciar ou parar as medições automáticas. Deve ser enviado um byte com tamanho do resto da mensagem, um byte com o número 10, e um byte (READY), indicando se deve iniciar ou não as medições. (1 = realizar medições / 0 = parar medições).

A estrutura da mensagem a ser enviada é:

BYTE 0	BYTE 1	BYTE 2
LENGTH	10	READY

A mensagem de resposta será uma mensagem de confirmação, descrita no tópico 2.1.

6. Exportação dos Dados:

Leitura dos dados diretamente do cartão SD (sistema de arquivos a definir);
Exportação dos dados para formato CSV;
Exportação dos dados para formato .xls;
Exportação dos dados para formato .xml;
Armazenar os dados em um SGBD;

7. Apresentação dos Dados:

Apresentar dados coletados da rede em tempo real, com gráficos e mapas identificando a posição dos nós;
Apresentar conjunto de dados extraídos do cartão SD;
Fitting de curvas;
Ferramentas de análise estatística.

8. Geração de Relatórios:

Gerar relatórios de dados por intervalo de tempo;
Gerar relatórios de dados por localização;
Gerar relatórios de dados por profundidade.

9. Status do Sistema:

Exibir status de memória do datalogger e da EEPROM da sonda;
Exibir carga restante da bateria;
Formatação do cartão SD e da memória não-volátil.

10. Configuração da Sonda:

Wizard de calibração dos sensores, com interface explicativa em forma de tutorial, requisitando as informações necessárias passo a passo;
Carregar e salvar os dados dos procedimentos de calibração;
Atualizar as constantes de calibração de um sensor e enviá-las à sonda;
Configurar o ID da sonda;
Configurar data e hora do sistema e da sonda
Habilitar/desabilitar sensores;
Especificar intervalos de medição para a aquisição de dados;
Configurar os diversos parâmetros do modem acústico.

11. Configuração do Software:

Configurar porta de comunicação, baud rate, e outros aspectos de comunicação serial;
Prover opções para intervalo de amostragem em tempo real e permitir ao usuário definir o intervalo de gravação dos dados no modo de aquisição normal, respeitando um limite de pior caso;
Configurar unidades de medida. Os valores fornecidos pela sonda estarão convencionados em uma unidade de medida padrão e caberá ao software prover regras de conversão em unidades opcionais;