

JULIO MITSUHIRO OBUCHI

Proposta de um Equipamento Telemedidor do Consumo e Nível de um
Reservatório de Água Aplicado em uma Estação do Metrô de São Paulo

São Paulo

2016

JULIO MITSUHIRO OBUCHI

Proposta de um Equipamento Telemedidor do Consumo e Nível de um
Reservatório de Água Aplicado em uma Estação do Metrô de São Paulo

Monografia apresentada para a conclusão
do Curso de Especialização em
Tecnologia Metroferroviária pela ESCOLA
POLITÉCNICA DA UNIVERSIDADE DE
SÃO PAULO.

São Paulo

2016

JULIO MITSUHIRO OBUCHI

Proposta de um Equipamento Telemedidor do Consumo e Nível de um
Reservatório de Água Aplicado em uma Estação do Metrô de São Paulo

Monografia apresentada para a conclusão
do Curso de Especialização em
Tecnologia Metroferroviária pela ESCOLA
POLITÉCNICA DA UNIVERSIDADE DE
SÃO PAULO.

Área de concentração:
Tecnologia Metroferroviária

Orientador:
Prof. Dr. Paulo Sérgio Cugnasca

São Paulo

2016

Obuchi, Julio Mitsuhiro

Proposta de um equipamento telemedidor do consumo e nível de um reservatório de água aplicado em uma estação do Metrô de São Paulo / J. M. Obuchi -- São Paulo, 2016.

41 p.

Monografia (Especialização em Tecnologia Metroferroviária) - Escola Politécnica da Universidade de São Paulo. PECE – Programa de Educação Continuada em Engenharia.

1. Telemetria I. Universidade de São Paulo. Escola Politécnica. PECE – Programa de Educação Continuada em Engenharia II.t.

AGRADECIMENTOS

Agradeço primeiramente à minha família por apoiar-me durante toda a realização deste trabalho.

Ao Metrô de São Paulo, por proporcionar a oportunidade de ampliar o meu desenvolvimento profissional com esta especialização.

Ao professor Paulo Sérgio Cugnasca, pela orientação e suporte na elaboração deste trabalho.

A todos os amigos e colegas que colaboraram direta ou indiretamente para o desenvolvimento e conclusão deste trabalho.

"A vida que a gente quer, depende do que a gente faz".

(Max Feffer)

RESUMO

O objetivo deste trabalho é propor um teledetector de consumo de água que permita a realização de leitura à distância, por telemetria, via comunicação IP em uma rede de dados *Ethernet*. Além de disponibilizar o valor do consumo de água, o detector também possui uma função adicional de monitorar os níveis mínimo e máximo de um reservatório. Diante da crise que houve no abastecimento de água na cidade de São Paulo, conhecer o consumo de água das instalações é uma ação imprescindível para a adoção de medidas corretivas e preventivas de economia. No caso da Companhia do Metropolitano de São Paulo existem vários pontos de abastecimento de água que estão distribuídos nas estações e prédios administrativos. Conhecendo-se os dados de consumo de forma rápida, pode-se detectar a existência de um consumo anormal e, com isso, combater os desperdícios. A solução proposta visa conceber um sistema de baixo custo de fabricação, permitindo a sua fácil implantação, manutenção e atualização dos seus elementos para uma estação típica do Metrô de São Paulo.

Palavras-chave: Telemetria. Medidor de consumo de água. Microcontroladores.

ABSTRACT

The objective of this work is to propose a water consumption telemeter. This device allows the users to measure the water consumption faraway, by telemetry, using an IP communication in Ethernet network. In addition to provide the water consumption value, the meter has the function to show the minimum and maximum level of the water reservoir. Against to the water supply crisis that happened in the São Paulo city, know the water consumption of the installations is essential to make the corrective and preventive economy actions. For the Companhia do Metropolitano de São Paulo there are many water supply points, which are located in stations and the administrative buildings. It is important to know quickly the data of consumption and to identify and combat the waste of water. The solution proposed here focus to make a cheaper system and to allow an easy way to implementation, repairing and updating of the components, considering a typical station of São Paulo's Metrô.

Keywords: Telemetry. Water consumption telemeter. Microcontrollers.

LISTA DE FIGURAS

Figura 1 – Elementos básicos na telemedição	13
Figura 2 – Diagrama de blocos do sistema de telemedição	19
Figura 3 – Módulo do microcontrolador Arduino Uno	20
Figura 4 – Módulo <i>Ethernet Shield</i> modelo W5100	21
Figura 5 – Módulo de relógio de tempo real DS1307	22
Figura 6 – Conexão do módulo de tempo real ao Arduino	23
Figura 7 – Sensor de fluxo	24
Figura 8 – Representação do resistor de <i>pull up</i>	26
Figura 9 – Fluxograma de funcionamento	27
Figura 10 – Monitoração do nível de água	28
Figura 11 – Placa de interface.....	30
Figura 12 – Montagem do protótipo	31
Figura 13 – Diagrama de blocos representando a ligação do medidor	32
Figura 14 – Resultado da medição no navegador.....	33
Figura 15 – Resultados obtidos com o medidor	36

LISTA DE TABELAS

Tabela 1 – Mapa das entradas e saídas utilizadas pelo Arduino.	25
Tabela 2 – Custos do protótipo	34
Tabela 3 – Resultados das medições de nível	35
Tabela 4 – Medições de consumo.....	36

LISTA DE ABREVIATURAS E SIGLAS

CHESS	Crise Hídrica, Estratégias e Soluções da Sabesp
CLP	Controlador Lógico Programável
CMSP	Companhia do Metropolitano de São Paulo - Metrô
EEPROM	<i>Electrically-Erasable Programmable Read-Only Memory</i>
GSM	<i>Global System for Mobile Communications</i>
IDE	<i>Integrated Development Environment</i>
IP	<i>Internet Protocol</i>
PWM	<i>Pulse Width Modulation</i>
RTC	<i>Real Time Clock</i>
Sabesp	Companhia de Saneamento Básico do Estado de São Paulo
SRAM	<i>Static Random Access Memory</i>

SUMÁRIO

1.	INTRODUÇÃO	9
1.1.	Justificativa	9
1.2.	Objetivos	10
1.3.	Estruturação do trabalho	10
2.	ESTADO DA ARTE – TELEMEDIÇÃO	12
2.1.	Elementos básicos de uma telemedicação	12
2.2.	Redes de comunicação em telemetria	13
2.3.	Aplicações da telemedicação	15
3.	PROPOSTA DE CONCEPÇÃO DO MEDIDOR	16
3.1.	Definições de escopo do projeto do medidor	16
3.2.	Análise de requisitos	16
3.3.	Arquitetura do <i>hardware</i>	18
3.4.	Sinais de entrada e saída de controle	25
3.5.	Funcionamento do medidor	26
3.6.	Elaboração do programa do microcontrolador	29
3.7.	Montagem do protótipo	29
3.8.	Testes do protótipo	31
3.9.	Custos do protótipo	33
4.	RESULTADOS	35
5.	CONCLUSÕES	37
6.	REFERÊNCIAS	38

1. INTRODUÇÃO

A proposta deste trabalho é a apresentação e o desenvolvimento de um dispositivo capaz de realizar a medição do consumo de água de um determinado ponto de abastecimento na estação do Metrô. O dispositivo deve ser capaz de transmitir o valor mensurado por telemetria, possibilitando assim a realização de leitura à distância. O meio de comunicação a ser utilizado é a rede *Ethernet*, que se mostra ser mais adequada e com maior disponibilidade para a aplicação dentro de uma estação do Metrô de São Paulo.

1.1. Justificativa

Diante do cenário de crise hídrica no abastecimento de água na região metropolitana de São Paulo iniciado em 2014, conhecer o consumo de água das instalações é uma ação imprescindível para a adoção de medidas corretivas e preventivas de economia.

O uso racional deste recurso natural é vital, não apenas devido às questões ambientais, mas também pelo agravamento do quadro de seca e a falta de chuvas para o armazenamento e recuperação dos reservatórios dos sistemas Cantareira e Alto Tietê (CHESS, 2015).

No caso da Companhia do Metropolitano de São Paulo – Metrô (CMSP) – existem vários pontos de abastecimento de água que estão distribuídos nas estações, pátios de manutenção e prédios administrativos. Monitorar manualmente o consumo da água e também o nível de seus reservatórios nos diversos pontos existentes é uma tarefa que requer tempo e a necessidade de deslocamento de um empregado para realizar a leitura, verificação e os registros dos dados.

Segundo os procedimentos operacionais do Metrô (CMSP, 2015), possuir a informação dos dados medidos de forma rápida é um passo necessário para o acompanhamento e o controle do consumo consciente de água. Dessa forma, este trabalho tem como motivação contribuir para a obtenção rápida das informações de consumo de água por meio de telemetria. Isso otimiza o trabalho de leitura do consumo e agiliza a adoção de estratégias de combate aos desperdícios de água dentro da CMSP.

1.2. Objetivos

O foco principal deste trabalho é a construção de um medidor de consumo de água e a transmissão dos dados por telemetria. Outra função agregada ao projeto do medidor é possibilitar também a monitoração do nível de água de um reservatório. Será uma opção do usuário do medidor decidir pela utilização ou não da monitoração do nível do reservatório, visto que o medidor pode estar instalado em um ponto do encanamento de água onde não existe uma caixa d'água.

Além disso, esta solução tem como meta ser desenvolvido com um baixo custo de fabricação, se comparada a outros dispositivos disponíveis no mercado. O medidor utiliza as tecnologias atualmente disponíveis, como os microcontroladores e os controladores lógicos programáveis (CLP). E com a vasta literatura técnica disponível na área eletrônica, permitir que seja uma opção viável de implantação, manutenção e atualização dos seus elementos, diante das soluções já existentes no mercado.

Uma vez que os dados de leitura do consumo de água sejam registrados, o medidor deve ser capaz de disponibilizá-los via rede, realizando a sua transmissão através de uma rede de comunicação com protocolo IP, compatível com a rede de comunicação interna do Metrô.

O usuário do sistema pode acompanhar, através de um navegador de *Internet*, o consumo de água do ponto monitorado, bem como saber o nível do reservatório de uma instalação em específico, identificando possíveis problemas e, dessa forma, tomar as ações corretivas que forem necessárias.

1.3. Estruturação do trabalho

Este trabalho foi dividido em algumas etapas. A primeira parte compreende a apresentação dos objetivos e motivos para a elaboração da proposta de um medidor de consumo de água aplicada a uma estação do Metrô.

O capítulo 2 aborda o estado da arte referente aos conceitos fundamentais de um sistema de telemetria.

No capítulo 3 estabelecem-se as premissas e os requisitos básicos adotados para a concepção do projeto do medidor. Também é descrito tecnicamente os materiais e procedimentos escolhidos para a construção do protótipo do medidor, assim como a estrutura de programação e o seu funcionamento. Para complementar os tópicos expostos, foram adicionadas os fundamentos teóricos essenciais para a compreensão do desenvolvimento técnico do trabalho, bem como foram descritos os motivos que justificam as escolhas do *hardware* empregado.

O capítulo 4 apresenta os resultados obtidos a partir da simulação e testes de um protótipo do medidor.

E finalmente, no capítulo 5, são apresentadas as conclusões obtidas neste trabalho, assim como as oportunidades de melhorias futuras do projeto.

2. ESTADO DA ARTE – TELEMEDIÇÃO

A telemedicação ou telemetria permite que uma determinada grandeza física seja mensurada remotamente, isto é, medida à distância e sem a necessidade do usuário estar presente diretamente no local de medição. A palavra telemetria tem origem grega, onde *tele* significa remoto e *metron* significa medida (Holanda, 2007).

Uma das grandes vantagens da telemedicação é obter de forma mais rápida e segura os dados necessários, quando comparada a uma leitura realizada manualmente no local. A telemetria permite, ainda, a redução de custos operacionais do monitoramento manual e também dos custos decorrentes quando uma anormalidade não é percebida rapidamente.

Dessa forma, a telemetria representa uma nova possibilidade de gestão para o monitoramento e controle de parâmetros de sistemas e processos (Tamaki, 2003).

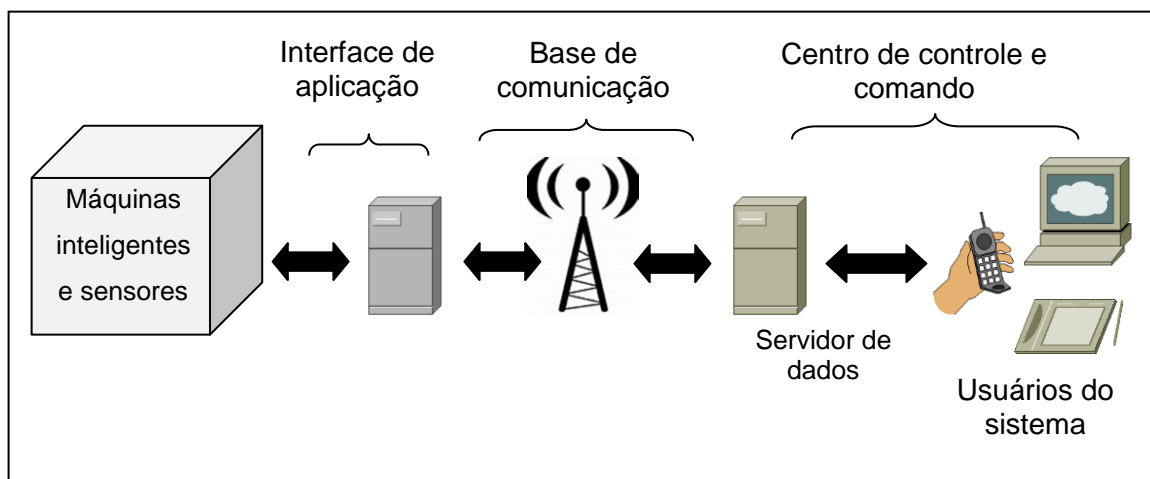
2.1. Elementos básicos de uma telemedicação

Conforme Bonde (2003), os sistemas de telemedicação são constituídos pelos seguintes componentes:

- Máquinas inteligentes e sensores: são os instrumentos e componentes que monitoram, controlam ou medem alguma variável local;
- Interface de aplicação: é o dispositivo que faz a ligação entre os dados dos sensores e a rede de comunicação dos dados;
- Base de comunicação: conhecido também por *backbone*, é o meio por onde trafega a informação dos sensores até um computador central;
- Centro de controle e comando: é o responsável por receber os dados transmitidos pelos sensores, onde a informação será processada e distribuída ou disponibilizada aos usuários finais.

A figura 1 representa de uma forma geral, os elementos básicos existentes num sistema de telemedicação e como ocorre o fluxo dos dados entre os sensores e máquinas até os clientes finais.

Figura 1 – Elementos básicos na telemedição



Fonte: adaptado de Bonde (2003)

Para implantar um sistema de telemedição, é preciso que tais elementos estejam presentes e desempenhem corretamente as suas funções. Sendo assim, o processo da telemetria inicia-se com a aquisição de dados. O sensor é o dispositivo capaz de converter uma grandeza física do ambiente – como a temperatura, a velocidade, a vazão e pressão, por exemplo – em um sinal elétrico mensurável. Esse sinal será então codificado e transmitido pela interface de aplicação, que pode ser, por exemplo, realizado por um controlador lógico programável (CLP). Em seguida, a informação é transmitida por uma rede de comunicação, que pode ser feita por meio físico (fios ou cabos de fibra óptica) ou através de tecnologias sem fio que utilizam ondas eletromagnéticas, como a radiofrequência ou em redes de telefonia celular, também conhecidas por GSM – *Global System for Mobile Communications*.

Por fim, uma central de controle informatizada recebe os dados que foram transmitidos, realiza sua decodificação e disponibiliza o resultado para o usuário final. Esse cliente pode acessar os dados de diversas formas, como em um terminal de computador, ou ainda por meio de aplicativos de *smartphone* ou *tablet*.

2.2. Redes de comunicação em telemetria

Existem diversos meios de comunicação para a transmissão de dados. Entre eles podem-se citar as micro-ondas, rádio privado, linha de telefonia, rede GSM, rede elétrica, fibra óptica, satélite e sistemas de redes proprietárias.

Segundo Alves (2011), a seleção de uma determinada rede de comunicação em telemetria é determinada em função da área de cobertura, dos custos de implantação, da disponibilidade e confiabilidade dos equipamentos da solução e da sua facilidade de integração. O peso de cada um desses fatores depende dos requisitos desejados pelo cliente final.

Segundo Bonde (2003), em geral, os sistemas de telemedição no Brasil dos setores de energia elétrica utilizam a rede elétrica como meio de comunicação. Já no restante do mundo é maior o uso da comunicação sem fio – *wireless* – como nos sistemas de radiofrequência ou pelas redes de telefonia celular (GSM). Como vantagem de uma rede GSM pode-se citar a sua rápida implantação, geralmente devida à existência de infraestrutura já pronta em muitos locais e pela facilidade de obter os dados de localizações com difícil acesso terrestre. Porém algumas desvantagens são a abrangência da área de cobertura do sinal, a necessidade de obtenção de licenças de órgãos reguladores de telecomunicações e a disponibilidade do sistema, que muitas vezes estão compartilhadas com outros serviços de telecomunicação, como a comunicação de voz.

Outra forma de transmissão de dados é o uso de sistemas cabeados com fio. A vantagem é a maior disponibilidade e confiabilidade no envio das informações, além de permitir altas taxas de transmissão. No entanto, a desvantagem é a necessidade de implantação de uma infraestrutura mais complexa.

Outra tecnologia de transmissão que vem sendo usada atualmente é o padrão *zigbee*. Segundo Scorzoni (2011), o padrão *zigbee* determina as especificações de dispositivos eletrônicos para a comunicação sem fio entre eles, permitindo um menor consumo e potência de operação. Embora o alcance da rede seja menor entre os dispositivos ligados numa rede *zigbee*, a comunicação pode ser repetida sucessivamente pelos diversos dispositivos da rede, retransmitindo a informação até que atinja o seu destino final.

De qualquer forma, para cada aplicação e necessidade de implantação de telemetria, devem-se analisar os requisitos necessários de abrangência, disponibilidade e custos. A escolha da melhor solução de comunicação dos dados depende da análise técnica e da melhor relação custo e benefício.

2.3. Aplicações da telemedicação

Os sistemas de telemetria são encontrados em diversas aplicações. Como exemplo de aplicação, podem-se citar os dados recolhidos por uma estação meteorológica. No setor energético, a telemedicação é aplicada para a medição de consumo de energia elétrica de uma unidade consumidora. No setor de transportes, a telemetria serve para o recolhimento de informações relativas, como a velocidade e posição da condução de um veículo ferroviário ao centro de controle operacional. Outro exemplo é no automobilismo esportivo, onde os engenheiros de uma equipe podem obter dados de velocidade, rotação do motor, temperatura do óleo de um carro de corrida. Isso permite que sejam realizados diagnósticos do comportamento do carro durante o decorrer da corrida.

Conforme é citado por Tamaki (2003), a telemedicação em conjunto com um sistema de gerenciamento de dados informatizados por *software*, possui a vantagem de realizar leituras em tempo real e com alta frequência de aquisição de dados. Isso possibilita o levantamento do perfil da variável mensurada, o levantamento de curvas características e a análise dos dados para a criação de parâmetros de controle, permitindo a detecção de anormalidades e a adoção de medidas preventivas ou corretivas que forem pertinentes.

No Brasil, conforme citado por Bonde (2003), os setores que mais utilizam as soluções de telemedicação são:

- Segurança: telemetria sem fio para comunicação dos sensores de alarmes voltados a segurança de equipamentos e patrimônios;
- Serviços públicos: medição de consumo de energia elétrica, água e gás de residências, comércios e indústrias; no caso, o setor de eletricidade é o que mais utiliza a telemetria.

3. PROPOSTA DE CONCEPÇÃO DO MEDIDOR

Para o desenvolvimento do medidor foi utilizada uma metodologia de prototipação, com a montagem de um modelo e realização de testes. Inicialmente foram definidas as premissas e condições iniciais de projeto. Com estes dados, foram descritos os requisitos para as escolhas dos componentes do *hardware* e o desenvolvimento do *software* do medidor.

3.1. Definições de escopo do projeto do medidor

Um passo importante para a execução deste projeto é a definição do escopo. Isso permite estabelecer as regras e a abrangência do projeto, possibilitando o controle das ações para atingir os objetivos propostos.

Para a construção do protótipo de medidor, foram definidos os seguintes parâmetros, bem como foram adotados algumas outras premissas iniciais do projeto:

- O medidor deve ser capaz de medir o consumo de água em litros;
- O projeto deve utilizar componentes de baixo custo;
- A arquitetura escolhida deve permitir facilidade para a montagem, elaboração do programa e testes;
- A medição deve ser realizada a distância, por telemetria;
- A comunicação utiliza a rede de dados *Ethernet*, para realizar a transmissão da informação, considerando que o medidor será aplicado em uma estação do Metrô;
- O medidor disponibiliza também outras funções adicionais, como a verificação dos níveis máximo e mínimo de um reservatório de água.

3.2. Análise de requisitos

Com base nas informações das premissas estabelecidas para o projeto, fez-se necessário a realização de uma análise prévia dos requisitos, isto é, a verificação e determinação dos parâmetros das variáveis de entrada e saída do sistema. Primeiramente, para possibilitar a automatização da leitura do consumo de água e transmissão dos dados por telemetria, foi decidido pela utilização de um

microcontrolador. Embora a sua programação tenha um maior nível de complexidade, o seu custo é menor quando comparado a um controlador lógico programável e também possui uma grande variedade de modelos e fabricantes disponíveis no mercado.

Com relação às variáveis de entrada do sistema, foi verificada a necessidade de obter um sinal de entrada, do tipo digital e compatível com o microcontrolador, que forneça o fluxo consumo de água em litros. Independente do modelo do sensor, a unidade de medida em litros pode ser obtida e convertida por programação.

Outra entrada de informação do tipo digital prevista para o telemedidor é a monitoração da condição do nível do reservatório de água. Neste caso, numa pesquisa realizada em campo, verificou-se que as chaves de nível existentes nos reservatórios podem ser utilizadas para esse propósito, dispensando a necessidade de instalação de outros sensores.

Em relação às variáveis de saída do sistema, a necessidade é a transmissão dos dados de consumo de água e situação do nível utilizando-se o protocolo de comunicação IP. Dessa forma, a comunicação dos dados utiliza a rede cabeada tipo *Ethernet*, que está ligada ao sistema de transmissão de dados da estação. Como os dados a serem transmitidos são simples, as velocidades e taxas de transmissão suportadas pela rede *Ethernet* são suficientes para o propósito do telemedidor.

Em termos de tempos de varredura e amostragem das variáveis, a leitura de consumo de água e monitoração do nível do reservatório não são atividades críticas. Dessa forma foi determinado, de forma empírica, um tempo de amostragem de 01 minuto para o telemedidor. Com isso, uma especificação de tempo de varredura menor que 10 ms ou 100 Hz é suficiente para atender as necessidades do projeto.

E com relação a aspectos não funcionais do projeto, é desejável que a disponibilidade do protótipo do telemedidor seja alta para que as informações estejam sempre disponíveis aos usuários finais do sistema.

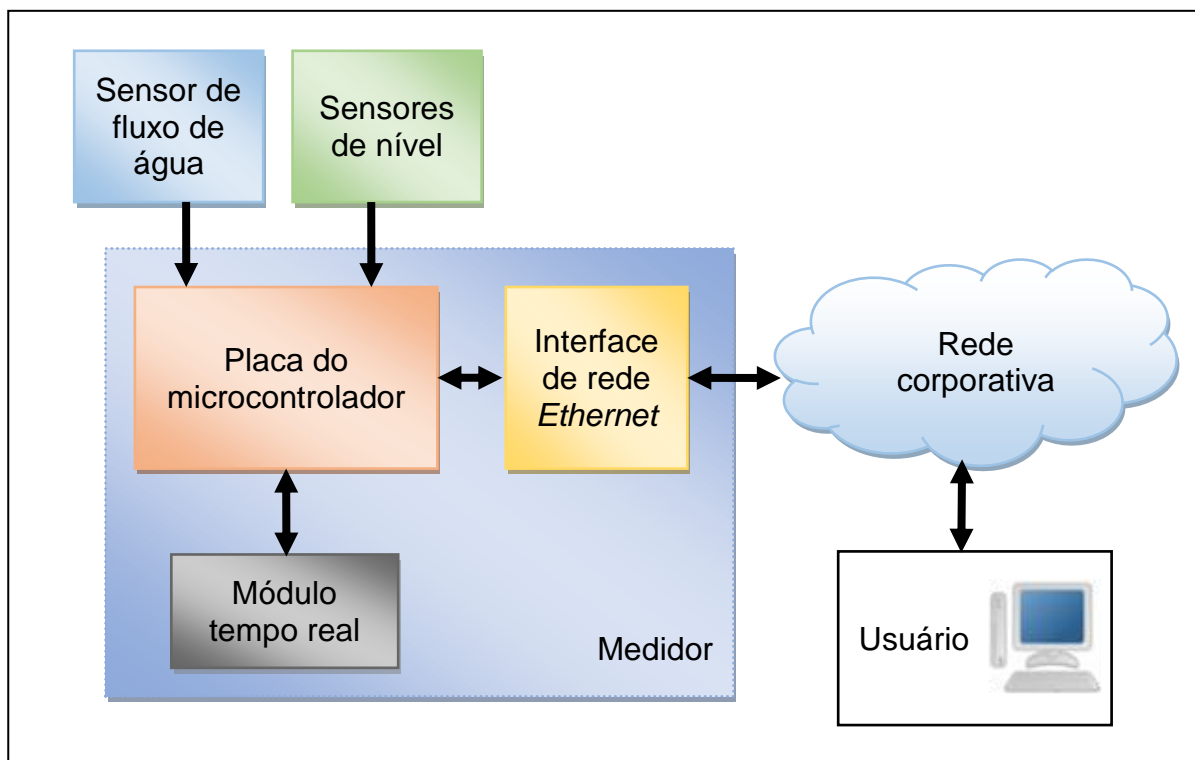
3.3. Arquitetura do *hardware*

Conforme os requisitos levantados, para o desenvolvimento do protótipo do telemedidor de consumo de água, foram definidos os componentes básicos do sistema, que são:

- Sensor de fluxo de água: é um transdutor que fornece um sinal digital na forma de pulsos, sobre a quantidade de água que passa pelo dispositivo de medição;
- Sensor de nível: é um sensor que possui um contato que muda de estado dependendo do nível de água que estiver ajustado;
- Placa de controle com microcontrolador: realiza a função de monitorar a leitura do sensor e realizar o processamento do sinal, enviando o resultado via rede de comunicação;
- Módulo de tempo real: é um componente que fornece ao sistema, os dados de data e hora atualizados. Este módulo possui uma bateria interna que preserva os dados mesmo com interrupção do fornecimento de energia elétrica;
- Placa de interface *Ethernet*: permite que o dado processado pelo microcontrolador seja transmitido via rede *Ethernet* para o usuário final.

O diagrama de blocos na figura 2 apresenta a arquitetura geral do *hardware* empregado, mostrando o medidor e suas interfaces de entrada e saída.

Figura 2 – Diagrama de blocos do sistema de telemedição



Fonte: do autor

3.3.1. Placa do microcontrolador

A área eletrônica evolui hoje com uma velocidade surpreendente e, constantemente, novos componentes são criados e desenvolvidos. Dentre os diversos recursos que existem atualmente, os microcontroladores estão cada vez mais acessíveis aos técnicos e engenheiros, facilitando o desenvolvimento de novos produtos e tecnologias. Segundo Souza (2003, p. 21), “poderíamos definir o microcontrolador como um pequeno componente eletrônico, dotado de uma inteligência programável, utilizado no controle de processos lógicos”.

A escolha do microcontrolador é parte importante para o desenvolvimento do protótipo do medidor de consumo de água.

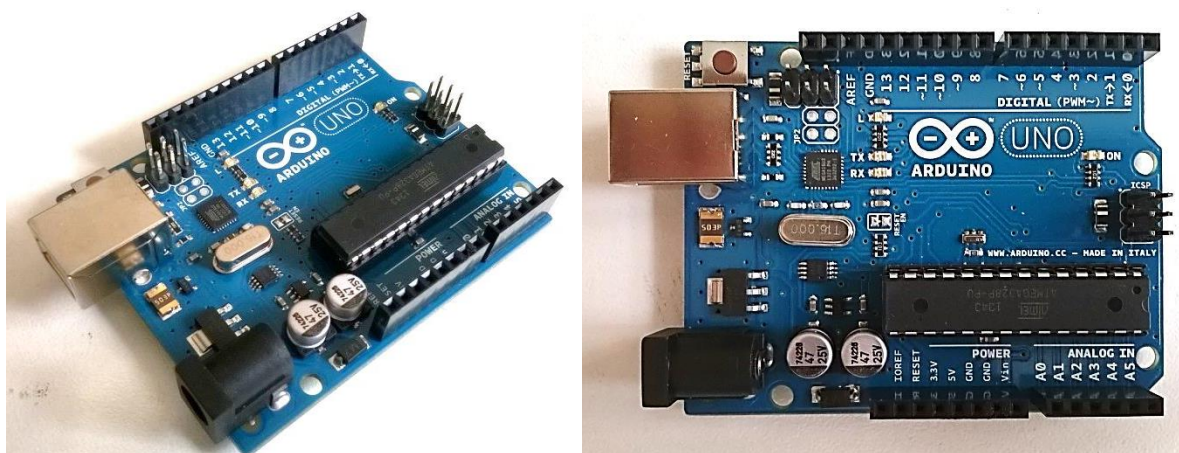
Dentre os diversos modelos de microcontroladores existentes no mercado, optou-se pela escolha de um modelo de baixo custo e com facilidade de aquisição e obtenção de literatura técnica. O modelo de microcontrolador adotado neste projeto foi o Arduino Uno rev3 que, além de atender aos requisitos identificados, possui uma

plataforma do tipo código aberto ou *open source* em inglês. Ou seja, esta plataforma é usada para o desenvolvimento de projetos que podem estar conectados a um computador. Outra característica é o seu *hardware* permitir a conexão de outros módulos adicionais para a execução de funções específicas, tornando o seu uso prático e versátil. Suas principais características técnicas são:

- Microcontrolador ATmega328P de 8 bits;
- Tensão de operação 5V;
- Tensão de entrada recomendada: 7 a 12V;
- Tensões limites de entrada: 6 a 20V;
- 14 portas de entrada ou saída configuráveis;
- 6 entradas/saídas digitais tipo PWM;
- 6 pinos de entrada analógica;
- Memória *flash* de 32kB;
- Memória SRAM de 2kB;
- Memória EEPROM de 1kB;
- *Clock* de 16 MHz.

A figura 3 mostra o aspecto físico do módulo do Arduino que foi adquirido para o desenvolvimento do protótipo do telemedidor.

Figura 3 – Módulo do microcontrolador Arduino Uno

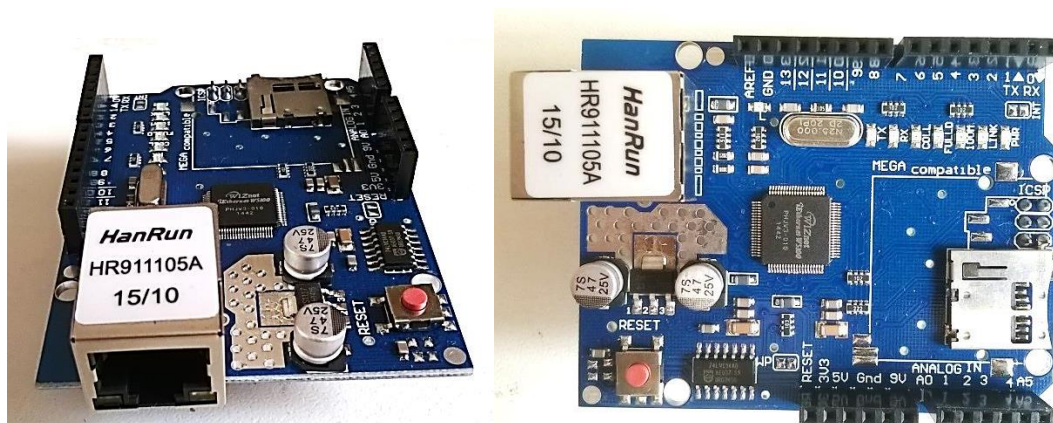


Fonte: do autor

3.3.2. Módulo de comunicação *Ethernet*

O padrão de comunicação estabelecido entre o medidor e o usuário foi o protocolo IP com rede tipo *Ethernet*. Dessa forma, dentre os componentes disponíveis no mercado, optou-se pela utilização do módulo de comunicação *Ethernet Shield* W5100, conforme mostra a Figura 4:

Figura 4 – Módulo *Ethernet Shield* modelo W5100



Fonte: do autor

Além da fácil aquisição, este dispositivo é desenvolvido e compatível com a plataforma Arduino. Isso facilita a sua montagem e integração com o sistema. Este tipo de módulo é também conhecido como *shield*. No mercado existem diversos modelos de *shields* que executam funções específicas e que são acoplados ao Arduino. Dessa maneira, pode-se aumentar a capacidade de aplicações do microcontrolador Arduino somente conectando uma placa à outra.

Cabe ressaltar que a escolha pela comunicação por rede *Ethernet* deve-se às vantagens apresentadas em relação a uma transmissão via radiofrequência dentro de uma estação do Metrô. Embora a rede *Ethernet* necessite de uma conexão física por cabo, uma transmissão por ondas de rádio ou GSM pode sofrer interferências das características físicas e construtivas do local e de disponibilidade ou variação de sinal.

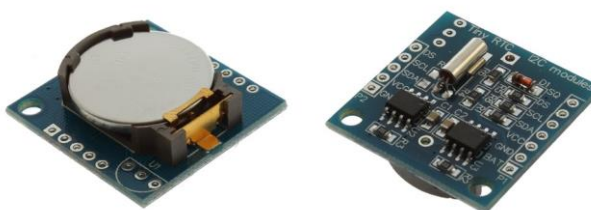
Como o medidor será aplicado em uma estação do Metrô, as transmissões de dados por ondas se mostram instáveis e indisponíveis em determinadas regiões do interior da estação. Muitos pontos de medição e reservatórios de água estão localizados em lugares sem cobertura de sinal de operadoras de telefonia celular – redes GSM.

Dessa forma, uma comunicação por rede *Ethernet* é a que garante uma maior disponibilidade para a transmissão dos dados do medidor dentro de uma estação do Metrô de São Paulo.

3.3.3. Módulo de relógio de tempo real (*Real time clock* - RTC)

Além da medição do consumo de água, o medidor precisa fornecer também a informação de data e hora juntamente com os dados de mensurados. Foi então identificada a necessidade de implantar um módulo com relógio de tempo real (*real time clock* ou RTC). O módulo escolhido foi o DS1307 que é capaz de fornecer dados de data e hora completas para o Arduino via comunicação serial. A bateria que pode ser conectada ao módulo, garante que as informações estejam disponíveis mesmo após a falta de alimentação elétrica do medidor. A Figura 5 mostra o aspecto físico deste módulo.

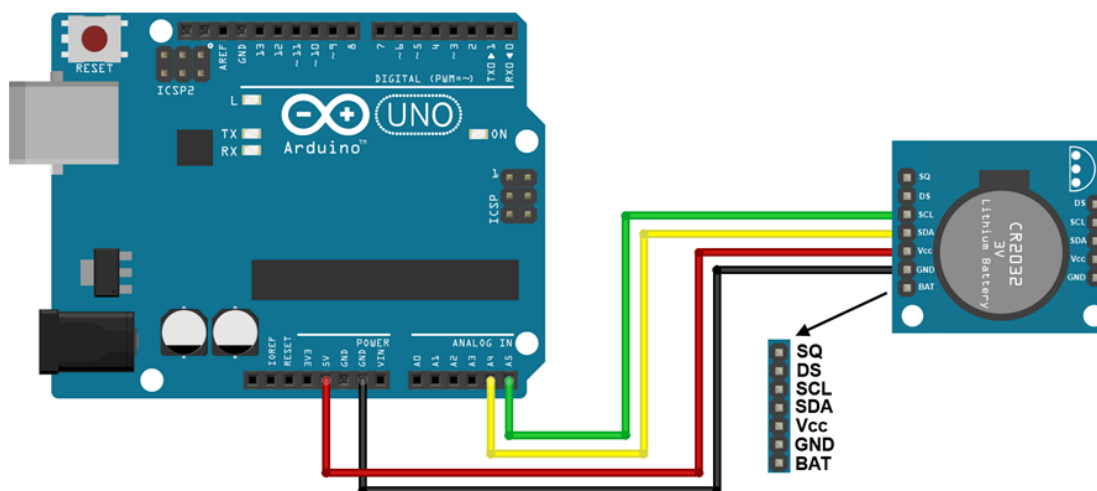
Figura 5 – Módulo de relógio de tempo real DS1307



Fonte: <<http://blog.filipeflop.com/modulos/relogio-rtc-ds1307-arduino.html>>

A conexão do módulo de tempo real com o Arduino é realizada da seguinte forma: dois cabos fornecem a alimentação de 5 Vcc ao módulo e mais dois cabos são conectados às entradas analógicas do microcontrolador, para a transmissão serial dos dados de data e hora, conforme está representado na Figura 6:

Figura 6 – Conexão do módulo de tempo real ao Arduino



Fonte: <<http://blog.filipeflop.com/modulos/relogio-rtc-ds1307-arduino.html>>

3.3.4. Definição de sensor de medição de fluxo de água e nível do reservatório

Para fins de montagem e teste do protótipo, foi utilizado um sensor de fluxo de água tipo velocímetro. Este sensor utiliza um transdutor de efeito *hall*, ou seja, o sensor não fica em contato direto com a água. A medição é realizada pela leitura de quantas voltas o mecanismo interno, conhecido como turbina, foi movimentado pela água que passou no seu interior. Devido à facilidade de aquisição de compra e da capacidade de vazão, o modelo escolhido para o protótipo foi o sensor YF-S201 fabricado pela empresa SEA®. Este sensor é compatível para aplicações que utilizam o Arduino, possuindo uma saída digital onde cada pulso gerado equivale a uma determinada quantidade de água que o atravessou. As principais características técnicas deste modelo de sensor são:

- Tensão de trabalho entre 5 a 18 Vcc;
- Vazão de trabalho de 01 a 30 litros por minuto;
- Pressão máxima de 2 Mpa;
- Temperatura de trabalho entre -25°C a +80°C;
- Pulsos por litro: 450;
- Frequência dos pulsos em Hz equivale a 7,5 vezes o fluxo em L/min.

Figura 7 – Sensor de fluxo



Fonte: do autor

Cabe destacar que a precisão deste modelo de sensor é da ordem de 10%. Existem outros modelos de sensores no mercado cuja precisão é maior e também com maior capacidade de vazão. A escolha do sensor mais adequado para uma aplicação no campo dependerá de um estudo complementar do perfil de consumo e da vazão do ramal a ser monitorado.

O medidor permite, ainda, que seja utilizado um sensor de maior capacidade de vazão, necessitando apenas de ajustes no *software* e que o pulso do sinal fornecido seja do tipo digital, permitindo, assim, a compatibilidade com o Arduino.

Para locais com maior demanda de consumo de água, a vazão e pressão podem superar as especificações do sensor de fluxo apresentado na figura 7. Dessa forma é recomendada a utilização de hidrômetros de maior capacidade, geralmente do tipo velocímetro e com possibilidade de acoplar sensores de medição remota, conforme os padrões estabelecidos pelas normas técnicas aplicáveis (SABESP, 2012).

Já para obter a informação do nível do reservatório de água, será considerado o uso da informação proveniente do sistema já existente no reservatório de uma estação típica do Metrô. Os reservatórios em geral possuem a chave de nível tipo boia que utilizados para o controle de bombas de água. A disponibilização de um contato normalmente aberto destas chaves é necessária para que o medidor informe a condição do nível do reservatório monitorado, ou seja, se o nível do reservatório está no mínimo ou se o nível está no máximo. Para fins de testes do protótipo, foram

adaptados ao circuito botões que simulam a condição do nível do reservatório de água.

3.4. Sinais de entrada e saída de controle

O microcontrolador Arduino possui portas configuráveis como entradas ou saídas, com opção de ser digital ou analógico. Para a montagem do protótipo, foi preciso estabelecer onde cada sinal é conectado ao módulo do Arduino. A Tabela 1 mostra a definição das entradas e saídas utilizadas pelo medidor. Alguns destes sinais são diretamente conectados entre as placas do microcontrolador e a interface *Ethernet*, otimizando, assim, a sua montagem. Outros sinais, como os sensores, foram conectados à placa do microcontrolador.

Tabela 1 – Mapa das entradas e saídas utilizadas pelo Arduino.

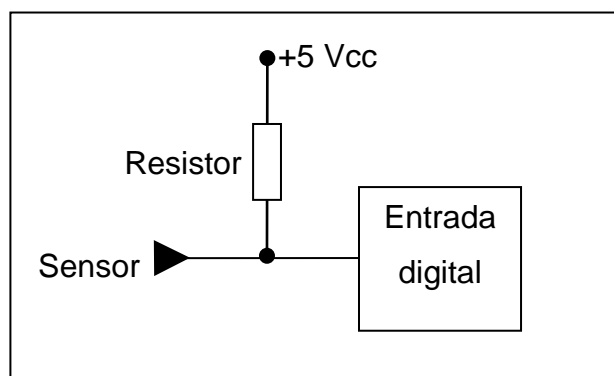
SINAIS DE ENTRADA		
Pino	Nome da entrada	Descrição
2	Sensor fluxo	Sinal de leitura do fluxo de água
5	Nível Mín.	Sinal do nível mínimo do reservatório
6	Nível Máx.	Sinal do nível máximo do reservatório
7	Botão Reset	Botão para reinicializar a contagem
A5	SCL	Sincronização na interface serial
A4	SDA	Transmite/recebe dados da interface
4	SS SD	Habilitação do cartão SD <i>card</i>
10	SS <i>Ethernet</i>	Habilitação do módulo <i>Ethernet</i>
11	MOSI	Canal comunicação dados <i>Ethernet</i>
12	MISO	Canal comunicação dados <i>Ethernet</i>
13	SCK	Sinal sincronização dados <i>Ethernet</i>

Fonte: do autor

Para evitar que possíveis flutuações de tensão provocassem uma leitura de sinais espúrios pelo medidor, foram adicionados resistores de *pull up* nas entradas digitais. Conforme McRoberts (2011, pag.66) a função deste resistor é garantir um nível lógico alto na entrada conectada. Um dos terminais do resistor é ligado à tensão positiva da fonte, enquanto o outro terminal é conectado na entrada digital do

microcontrolador. Dessa forma, enquanto que o sinal proveniente do sensor não estiver com nível lógico zero, a entrada digital é mantida com nível lógico alto. A Figura 8 representa a conexão do resistor de *pull up*:

Figura 8 – Representação do resistor de *pull up*

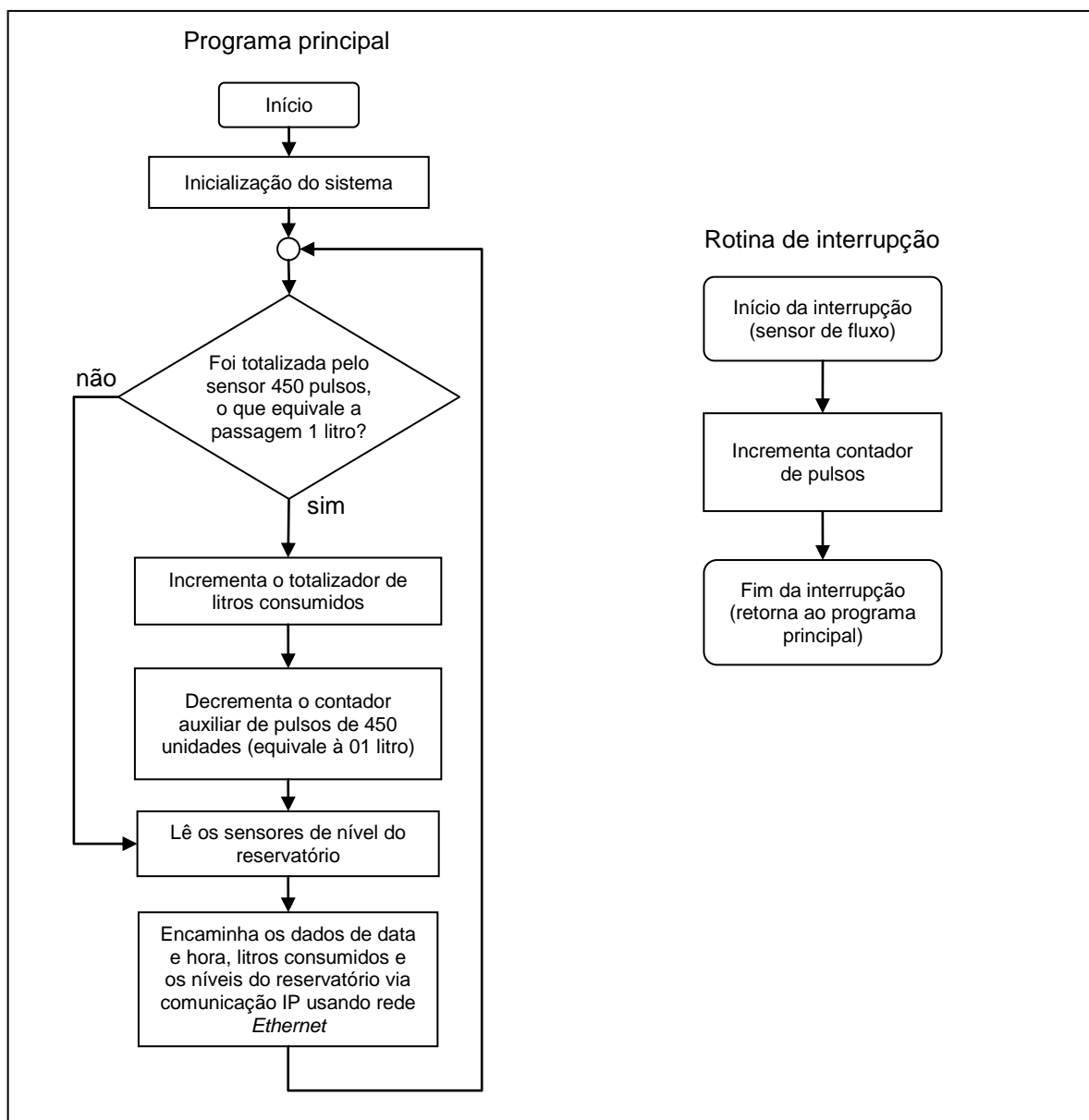


Fonte: do autor

3.5. Funcionamento do medidor

Para estabelecer como o medidor deve se comportar, foi elaborado um fluxograma que detalha o funcionamento básico do circuito, ou seja, como o medidor fará a leitura das informações dos sensores, o seu tratamento matemático e a transmissão dos dados. A Figura 9 mostra o fluxograma elaborado do medidor, que servirá de referência para a criação do programa (*software*) a ser gravado no microcontrolador:

Figura 9 – Fluxograma de funcionamento



Fonte: do autor.

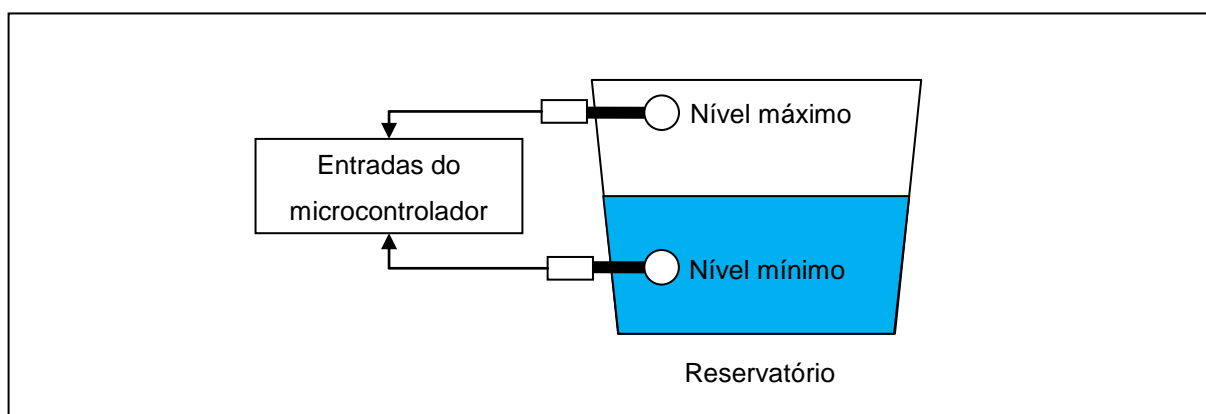
Primeiramente, foi idealizado que o medidor monitore a leitura de pulsos proveniente de um sensor de fluxo de água. Para o tratamento deste sinal, é utilizada uma rotina de interrupção do microcontrolador. Uma vez que isso ocorra, é incrementado um contador interno que registra a quantidade de pulsos registrados. Dessa forma, sempre que houver um evento de um pulso do sensor, o programa interrompe momentaneamente a tarefa que estiver em execução para que a rotina de interrupção seja executada. Isso permite garantir que nenhum registro de leitura dos

pulsos seja perdido. Uma vez que a rotina de interrupção seja concluída, o programa volta a ser executado a partir do ponto em que foi interrompido.

O programa principal, por sua vez, verifica sempre se uma determinada quantidade de pulsos equivalente a 1 litro de água foi acumulado. No caso do protótipo montado, foi utilizado o valor de 450 pulsos para representar o consumo de 1 litro de água. Esta informação foi retirada das características técnicas do sensor e deverá ser ajustada caso haja uma mudança do modelo de sensor utilizado. Uma vez que 450 pulsos sejam registrados, um contador com a função de totalizar a quantidade de litros consumidos é incrementado, registrando assim o valor de 1 litro. Por outro lado, o registrador da quantidade de pulsos é decrementado em 450, de forma que o programa aguarde que os pulsos restantes acumulem novamente um valor igual ou maior a 450 para que um novo litro seja totalizado ao consumo.

Na sequência, o programa continua a execução verificando os sensores de nível de água do reservatório. São monitorados os estados lógicos de duas entradas digitais do microcontrolador conectados aos sensores de nível mínimo e máximo. Quando os sensores de nível mínimo e máximo não estão ativados, o medidor sinaliza a condição de nível baixo do reservatório. Quando o sensor de nível de mínimo está ativado e o de nível máximo continua inativo, o reservatório está com o nível dentro do normal. E caso os dois sensores estejam ativados, o reservatório está com um nível acima do normal. A figura 10 representa um reservatório com os sensores de nível conectados.

Figura 10 – Monitoração do nível de água



Fonte: do autor.

Por fim, uma rotina de transmissão dos dados inicia uma comunicação IP via rede *Ethernet*. Dessa forma, o medidor é capaz de disponibilizar na rede as informações de data, hora, consumo em litros e a condição do nível do reservatório de água. O usuário do sistema deve, através de um navegador de *Internet*, acessar o respectivo endereço IP do medidor e assim obter os dados que foram mensurados e transmitidos.

Após o envio da informação, o microcontrolador fará uma nova rotina de leitura e repetirá o processo novamente. O ciclo de execução da rotina foi empiricamente ajustado para 05 segundos, de forma a permitir a realização dos testes do protótipo. Este tempo pode ser ajustado dentro do programa elaborado para o microcontrolador, de forma atender ao requisito especificado de 01 minuto apresentado no item 3.2.

3.6. Elaboração do programa do microcontrolador

O programa elaborado para este projeto utiliza como base o fluxograma detalhado no tópico anterior. Dessa forma, foi criado o algoritmo do programa em linguagem C no ambiente de desenvolvimento integrado (*Integrated Development Environment – IDE*) do Arduino.

Por ser uma plataforma de código aberto, o *software* está disponível gratuitamente na página oficial do Arduino na *Internet*. O código fonte desenvolvido para este projeto não está exibido neste capítulo, devido a sua extensão. Porém, ele está disponível na sua totalidade no anexo A, com comentários em várias das suas linhas de programação. Como o objetivo deste trabalho não é explicar com detalhes os conceitos e comandos da linguagem de programação, sugere-se a leitura de uma bibliografia complementar com relação a este assunto.

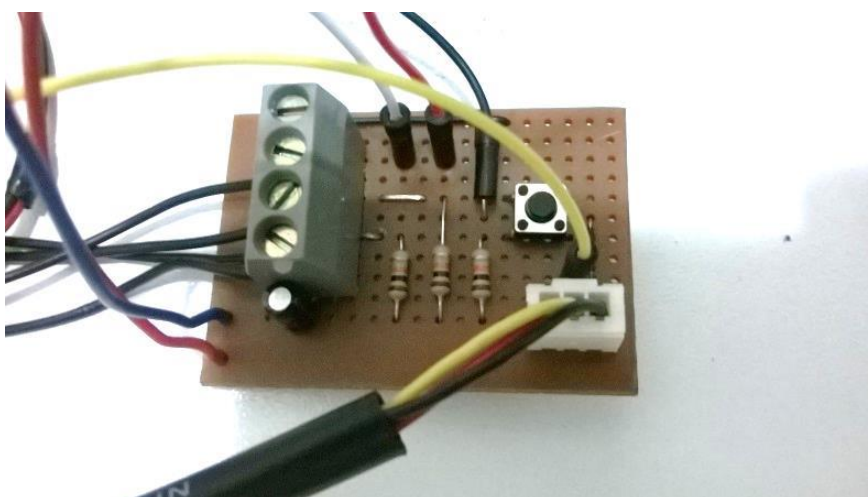
3.7. Montagem do protótipo

A montagem do protótipo foi realizada utilizando-se os componentes básicos previstos e improvisando ou adaptando outros elementos que estariam disponíveis somente no local de instalação do medidor, como no caso das chaves de nível.

As placas do microcontrolador Arduino e do módulo de comunicação *Ethernet* permitem uma conexão direta, bastando encaixar uma placa sobre a outra. Este conjunto de placas foi acomodado e fixado dentro de uma caixa acrílica própria que oferece uma proteção mecânica e, ainda, garante a isolação elétrica dos terminais expostos. Já o módulo de tempo real foi fixado na mesma caixa e os seus cabos foram conectados por meio de solda à placa do microcontrolador.

Como fonte de alimentação, foi utilizada uma bateria de 9 Volts conectada à placa do microcontrolador. Foi necessária, ainda, a confecção de uma placa de interface, contendo os resistores de *pull up* e o botão de *reset*, permitindo, assim, a ligação dos sensores às entradas do microcontrolador. A figura 11, a seguir, mostra a placa de interface.

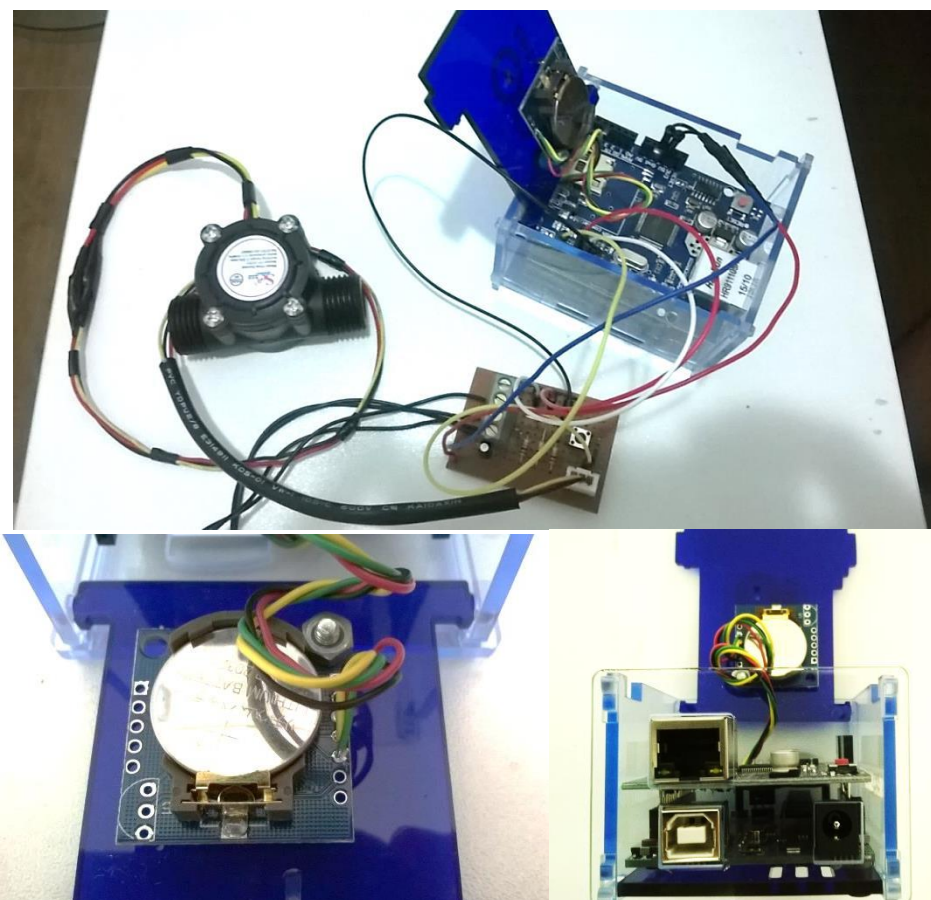
Figura 11 – Placa de interface



Fonte: do autor

Em seguida foram conectados todos os dispositivos ao medidor. A figura 12 ilustra o resultado final da montagem de todos os componentes do protótipo do medidor.

Figura 12 – Montagem do protótipo



Fonte: do autor

Uma vez que o protótipo foi fisicamente montado, a próxima etapa foi a gravação do programa dentro da memória do microcontrolador. Para isso, foi utilizado o mesmo *software* IDE de desenvolvimento do Arduino. Através de um cabo conectado entre o microcontrolador e o computador, é realizado o carregamento e gravação do programa. Realizado esta etapa, o medidor está apto para a realização dos testes.

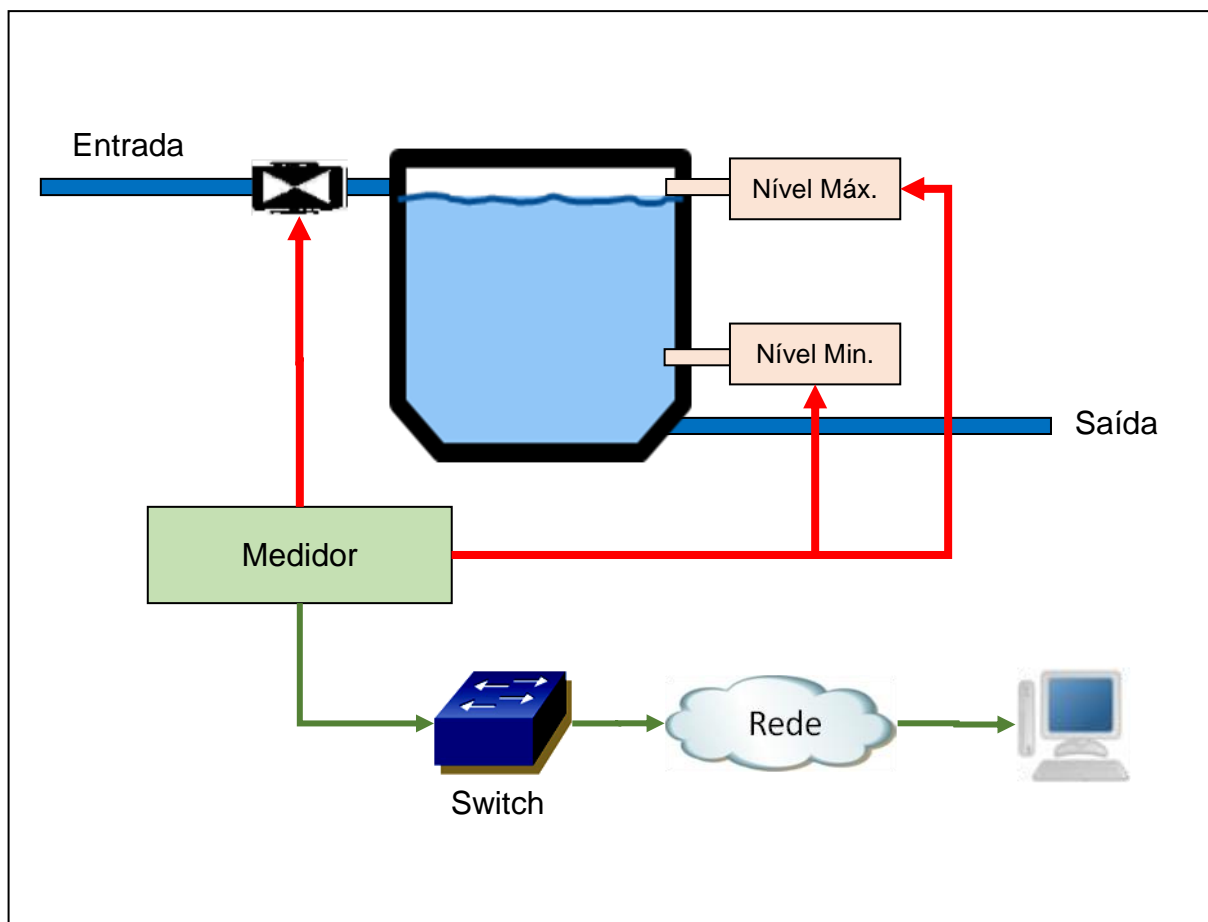
3.8. Testes do protótipo

Os testes de validação do funcionamento do protótipo foram realizados em bancada. A metodologia utilizada para a execução dos testes foi a simulação de passagem de água pelo sensor de fluxo e da condição dos estados dos sensores de nível.

Para o teste de comunicação foi conectado o medidor em uma rede doméstica através de uma porta *Ethernet* de um roteador com endereçamento IP. A figura 13 mostra, de uma forma simplificada, as ligações do medidor dentro de um sistema de

abastecimento da estação. Este sistema é, basicamente, composto de um reservatório e de um ramal de entrada e outro de saída da água.

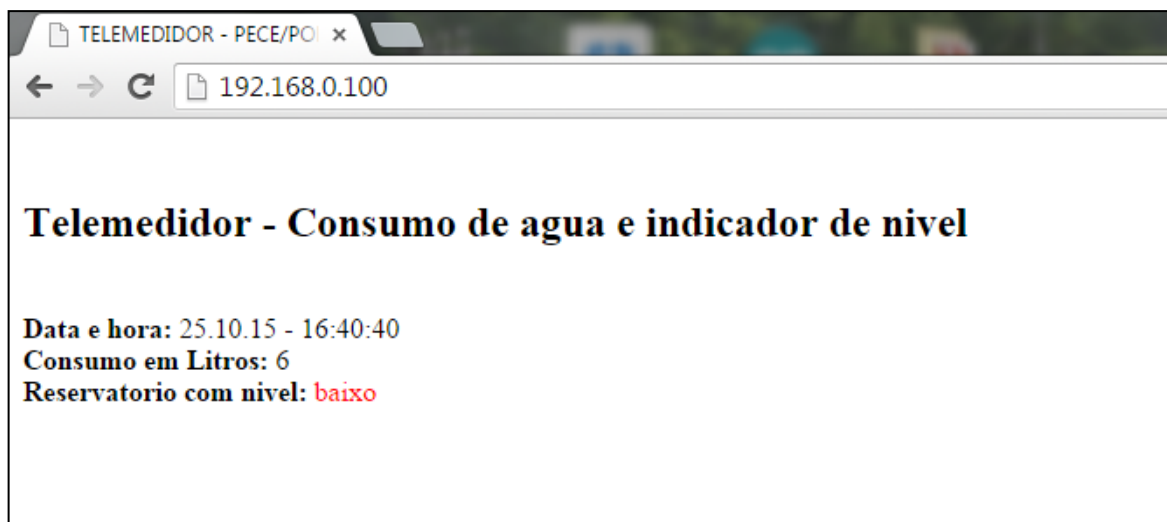
Figura 13 – Diagrama de blocos representando a ligação do medidor



Fonte: do autor

Os testes mostraram que o protótipo funcionou conforme o esperado. Foi realizada a leitura dos pulsos do sensor de fluxo e o consumo calculado foi disponibilizado via rede, através de um endereço IP. A figura 14 mostra uma tela obtida a partir de um computador conectado na mesma rede. Nesta tela é possível verificar, primeiramente, qual o dia e horário registrado pelo medidor. O consumo de água registrado atualmente é fornecido em litros. E, finalmente, é exibido qual a situação do reservatório de água.

Figura 14 – Resultado da medição no navegador



Fonte: do autor

Portanto, as informações recebidas via rede *Ethernet* do medidor foram as seguintes:

- Data e horário da medição;
- Consumo de água em litros;
- Indicação do nível do reservatório, onde tem-se as indicações de **baixo**, **normal** e **alto**, representando o estado do volume de água.

Para se determinar qual foi o consumo de água em um determinado período, como por exemplo o consumo diário, é necessário que as informações sejam guardadas em um banco de dados e posteriormente analisados com o auxílio de outro *software* de apoio.

O capítulo 4 descreve com mais detalhes os resultados obtidos dos testes realizados.

3.9. Custos do protótipo

Os custos para a montagem do protótipo foram determinados somando-se os valores de compra dos principais componentes utilizados na montagem física. Para alguns itens, foi estimado um custo aproximado. Entre esse materiais cujo valor foi estimado estão os fios, resistores, botões etc.

Já a despesa referente à mão de obra utilizada para o projeto foi desprezada por ser tratar do objetivo deste trabalho. Quanto a utilização de ferramentas e instrumentos, como por exemplo o alicate e o multímetro, estes também foram desconsiderados do cálculo de custo final do protótipo, pois também faz parte do desenvolvimento deste trabalho. A tabela 2 abaixo demonstra o custo do *hardware* utilizado para a montagem do protótipo:

Tabela 2 – Custos do protótipo

Quantidade	Descrição	Valor Unitário	Total
01	Modulo <i>Ethernet Shield</i> para Arduino	R\$ 52,00	R\$ 52,00
01	Arduino Uno R3 com cabo USB	R\$ 38,00	R\$ 38,00
01	Módulo RTC 1307	R\$ 20,00	R\$ 20,00
01	Bateria 9V	R\$ 16,00	R\$ 16,00
01	Sensor de fluxo ½"	R\$ 40,00	R\$ 40,00
Diversos	Resistores, fios, fita isolante, botões, conectores	R\$ 30,00	R\$ 30,00
Custo total			R\$ 196,00

Fonte: do autor

Conforme os dados da tabela, o custo calculado de material para o protótipo foi de R\$ 196,00. Porém, este valor pode sofrer variações de acordo com o custo dos componentes e da quantidade de medidores a serem produzidos.

De uma forma geral, pode-se considerar que o custo apresentado foi relativamente baixo. A justificativa para isso foi a utilização de componentes comumente encontrados no mercado.

4. RESULTADOS

Este capítulo descreve quais foram os resultados obtidos dos ensaios e testes realizados em bancada pelo medidor, que permaneceu conectado à uma rede *Ethernet*, enquanto os resultados eram lidos através de um computador.

A primeira verificação foi identificar a condição do nível de um reservatório fictício de água. As entradas digitais onde são conectadas as informações dos sensores de nível foram manualmente simuladas, de forma a obter os seguintes resultados da Tabela 3 a seguir:

Tabela 3 – Resultados das medições de nível

Situação do reservatório	Reservatório com nível baixo	Reservatório com nível normal	Reservatório com nível alto
			
Resultado esperado	Nível baixo	Nível normal	Nível alto
Resultado obtido			

Fonte: do autor

Os dados mostram que o medidor informou ao usuário, via comunicação de rede, qual era a condição do reservatório. A combinação do estado lógico dos sensores de níveis indicou a condição de nível baixo, normal ou alto.

Com relação ao teste de medição de consumo, foi realizada uma simulação da passagem de um volume previamente determinado de água no sensor de fluxo.

Foi despejada por meio de um recipiente com graduação uma quantidade de água de forma a obter um valor exato de medição em litros. Os valores obtidos de uma série de 10 medições experimentais foram:

Tabela 4 – Medições de consumo

Número da amostra	Valor do medidor em litros	Volume real em litros	Erro de leitura em %
1	2	2,1	5%
2	2	2,0	0%
3	2	2,0	0%
4	2	2,1	5%
5	2	2,2	10%
6	6	6,1	2%
7	6	6,3	5%
8	6	6,1	2%
9	10	10,3	3%
10	10	10,5	5%

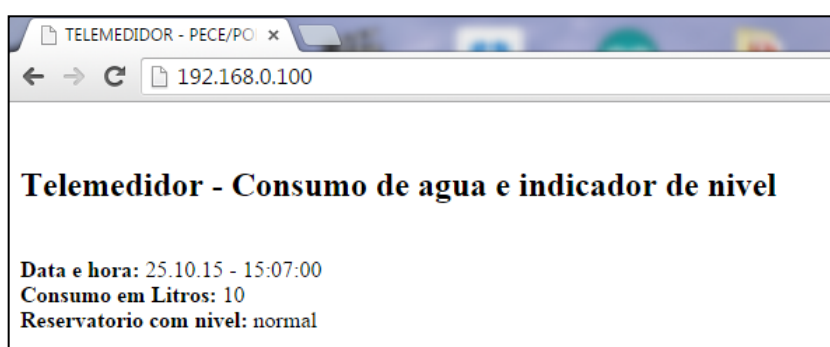
Fonte: do autor

Conforme os dados da tabela 4, o medidor mediu o consumo de água dentro da capacidade esperada. A precisão do sensor de fluxo utilizado para este protótipo possui uma margem de 10% de precisão, o que justifica o medidor apresentar o erro de leitura calculado na tabela.

Uma observação a ser feita é que a precisão do volume real em litros foi de 100 ml, pois o recipiente utilizado para medição possuía graduações de 100 ml.

A figura 15 mostra os resultados lidos com o medidor:

Figura 15 – Resultados obtidos com o medidor



Fonte: do autor

5. CONCLUSÕES

Com base nos resultados obtidos, pode-se afirmar que o protótipo do telemedidor mediu o consumo de água e transmitiu os dados via rede cabeada *Ethernet*, caracterizando assim a medição à distancia, por telemetria. Outra funcionalidade que foi testada e comprovada do telemedidor foi a monitoração da condição do nível de água de um reservatório.

Dessa forma, o usuário do sistema pode obter rapidamente os dados do campo, através um navegador da *Internet*, o que facilita a análise das informações e a tomada de decisões.

Com relação ao custo de montagem do protótipo, pode-se considerar que o valor foi baixo. Isso se deve a utilização de um microcontrolador e do uso de componentes comumente encontrados no mercado. Mas por outro lado, deve-se considerar que outros custos não foram agregados ao projeto do protótipo, como a utilização de ferramentas e mão de obra própria.

No entanto, ficou faltando uma avaliação do funcionamento do protótipo do telemedidor dentro das instalações de uma estação do Metrô de São Paulo. Somente assim é que poderá ser avaliada a aplicabilidade do medidor e, também, a verificação de outros requisitos do equipamento, como a sua confiabilidade, disponibilidade e robustez.

Além disso, sugere-se que o medidor seja integrado à um sistema informatizado, de forma a gravar os dados da telemetria em um servidor na rede administrativa do Metrô, e permitir assim que os dados sejam analisados de forma gerencial, ou seja, com elaboração de gráficos e relatórios de consumo por períodos de tempo através de um *software* próprio.

6. REFERÊNCIAS

ALVES, Jéssica Martins da Cruz Souza. **Telemedicação: As Telecomunicações Impulsionando a Energia**. 18/04/2011: Teleco – Inteligência em Telecomunicações. Disponível em <<http://www.teleco.com.br/tutoriais/tutorialtelemed/default.asp>>. Acesso em 19 set. 2015.

BONDE, Ian. **Mercado Brasileiro de Telemetria**. 27/10/2003: Teleco – Inteligência em Telecomunicações. Disponível em <<http://www.teleco.com.br/tutoriais/tutorialmtelemetria/default.asp>>. Acesso em 19 set. 2015.

COMPANHIA DE SANEAMENTO BÁSICO DO ESTADO DE SÃO PAULO. **Documento CHES- Crise Hídrica, Estratégias e Soluções da Sabesp**, São Paulo, 30 de abril 2015, disponível em <www.sabesp.com.br>. Acesso em: set. 2015.

_____. **NORMA TÉCNICA SABESP NTS 181**. São Paulo. Novembro de 2012. Revisão 3. Disponível em <<http://www2.sabesp.com.br/normas/nts/NTS181.pdf>>. Acesso em 27 jul. 2015.

COMPANHIA DO METROPOLITANO DE SÃO PAULO. **Procedimento Operacional 13-305A02: Acompanhamento do consumo de água dos postos operativos**. Companhia do Metropolitano de São Paulo, 30 jul. 2015, revisão 9.

HOLANDA, Marcos Antônio Arruda Guerra de. **Medição individualizada em edifícios residenciais: controle e redução do consumo de água potável**. Recife, 2007. 135 p. Dissertação (Mestrado) – Universidade Católica de Pernambuco. Pró-reitoria de Ensino, Pesquisa e Extensão, 2007.

MCROBERTS, Michael. **Arduino básico**. 1º Edição, São Paulo: Novatec Editora, 2011.

SCORZINI, Fernando. **Desenvolvimento de um sistema de controle híbrido utilizando CAN e Zigbee**. São Carlos, 2011. 56 p. Monografia (Graduação) – Escola de Engenharia de São Carlos da Universidade de São Paulo, 2011.

SOUZA, David José de. **Desbravando o PIC: ampliado e atualizado para PIC 16F628A**. 6ª ed. São Paulo: Érica, 2003.

TAMAKI, Humberto Oyamada. **A medição setorizada como instrumento de gestão da demanda de água em sistemas prediais – estudo de caso: programa de uso racional da água da Universidade de São Paulo**. São Paulo, 2003. 151 p. Dissertação (Mestrado) – Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Construção Civil, 2003.

ANEXO A

Este anexo apresenta com detalhes o código do programa elaborado no *software* de desenvolvimento (IDE) do Arduino. Os comentários foram adicionados ao longo do programa para facilitar a sua compreensão.

```
//*****
// Carrega as bibliotecas necessárias
//*****
#include <SPI.h>      //biblioteca para protocolo de comunicacao Serial Peripheral Interface
#include <Ethernet.h> //biblioteca para comandos Ethernet
#include <DS1307.h>   //biblioteca para comandos do relógio tempo real

//*****
// Definicao de enderecos das entradas e saidas
//*****
// A linha abaixo permite que voce defina o endereço físico (MAC ADDRESS)
// da placa de rede
byte mac[] = { 0x00, 0x1E, 0xC9, 0xF9, 0xA9, 0x66 };
byte Int_Sensor = 0; // 0 = pin 2; 1 = pin 3 define o pino 2 como interrupcao
byte Sensor = 2;    // define o pino 2 como sinal do sensor
byte Nivel_Min = 5; // define pino 5 como retorno de nivel minimo
byte Nivel_Max = 6; // define pino 6 como retorno de nivel minimo
byte Botao_Reset = 7; // define pino 7 como botão de reset da contagem
DS1307 rtc(A4, A5); // define que o Modulo RTC DS1307 ligado as portas analógicas A4 e A5

//*****
// Os valores abaixo definem o endereço IP, gateway e máscara. Precisa configurar de acordo
// com a rede conectada. Em geral o IP 192.168.0.xxx eh comum de ser utilizado
//*****
IPAddress ip(192,168,0,100); //Define o endereço IP
IPAddress gateway(192,168,0,1); //Define o gateway
IPAddress subnet(255, 255, 255, 0); //Define a máscara de rede
EthernetServer server(80); //Inicializa biblioteca ethernet com configurações IP fornecidas

//*****
// Definicao de variaveis gerais do programa
//*****
int Contagem; //variavel para uso em interrupcao
unsigned long Litros; //2 bytes variando 0 a 65535
int TotalLitros; //4 bytes variando 0 a 4294967295
int pulsos;

//*****
// Configuracoes gerais do sistema. A rotina "setup" é executada uma unica vez no inicio do programa
//*****
void setup()
{
    // Definicao da configuracao dos pinos de entrada
    pinMode(Botao_Reset, INPUT); // define pino como entrada
    pinMode(Sensor, INPUT);
    pinMode(Nivel_Min, INPUT);
    pinMode(Nivel_Max, INPUT);

    // Inicializa a conexao ethernet e o servidor web na porta 80
    Ethernet.begin(mac, ip);
    server.begin();
}
```

```

// Reinicializa as variaveis gerais
Contagem = 0;
Litros = 0;
TotalLitros = 0;

// Configura a interrupcao do pino 2, chama a subrotina
// Contagens e le a borda de descida do sinal
attachInterrupt(Int_Sensor, Contagens, FALLING);

sei();

rtc.halt(false);    // Aciona o relógio RTC

//*****
// As linhas de código abaixo servem para ajustar a data e hora do módulo
// RTC e devem ser omitidas ou comentadas após a primeira utilização
//*****
// rtc.setDOW(TUESDAY);    //Define o dia da semana
// rtc.setTime(10, 51, 0);    //Define o horário
// rtc.setDate(25, 10, 2015); //Define o dia, mês e ano

//*****
// Definições do pino SQW/Out do módulo RTC
//*****
rtc.setSQWRate(SQW_RATE_1);
rtc.enableSQW(true);
}

//*****
// Programa principal que fica executando em loop
//*****
void loop()
{
    // Verifica o Botão_Reset e zera a contagem total
    if (digitalRead(Botao_Reset) == LOW) {
        TotalLitros = 0; //zera a contagem
    }

    cli();
    pulsos = Contagem;
    sei();

    if (pulsos >= 450) {
        TotalLitros++;
        cli();
        Contagem = Contagem - 450; //decrementa do contador o consumo de 1 litro
        sei(); // habilita interrupção
    }

    // rotina para mandar dados
    EthernetClient client = server.available();
    if (client) {
        Serial.println("Novo Cliente");
        boolean currentLineIsBlank = true;
        while (client.connected()) {
            if (client.available()) {
                char c = client.read();
                Serial.write(c);
                if (c == '\n' && currentLineIsBlank) {

```

```

client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println("Connection: close");
client.println("Refresh: 5"); //o valor 5 determina o ciclo em segundos para a rotina
client.println();
client.println("<!DOCTYPE HTML>");
client.println("<html>");
client.println("<head><title> TELEMEDIDOR - PECE/POLI 2015 </title></head>");
client.println("<br />");
client.println("<B><H2>Telemididor - Consumo de agua e indicador de nivel</H2></B>");
client.println("<br />");
client.println("<B>Data e hora: </B>");
client.print(rtc.getDateStr(FORMAT_SHORT));
client.print(" - ");
client.print(rtc.getTimeStr());
client.println("<br />");
client.println("<B>Consumo em Litros: </B>");
client.print(TotalLitros);
client.println("<br />");
client.println("<B>Reservatorio com nivel: </B>");

    if ((digitalRead(Nivel_Min) == LOW)&&(digitalRead(Nivel_Max) == LOW)) {
client.println("<FONT color=\"red\"> baixo </FONT>");
    }
    if ((digitalRead(Nivel_Min) == HIGH)&&(digitalRead(Nivel_Max) == LOW)) {
client.println("normal");
    }
    if ((digitalRead(Nivel_Min) == HIGH)&&(digitalRead(Nivel_Max) == HIGH)) {
client.println("<FONT color=\"red\"> alto </FONT>");
    }

    client.println("<br />");
    client.println("</html>");
    break;
}
if (c == '\n') {
    currentLineIsBlank = true;
}
else if (c != '\r') {
    currentLineIsBlank = false;
}
}
}
delay(10);
client.stop();
Serial.println("client disconnected");
}
}

//*****
// Tratamento para a interrupcao. Quando o sensor registrar um pulso, a variavel
// Contagem é incrementada.
//*****
void Contagens()
{
    Contagem++;
}

```