



UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
SISTEMA DE AQUISIÇÃO DE DADOS E INTERFACE

RELATÓRIO: ETAPA 3

"Aquisição de Dados de um Aquário"

Danilo Gomes de Andrade - 120111433

Henrique Dantas Silva – 116110287

João Victor Rodrigues Guimarães – 115210493

João Victor Rodrigues Ramalho - 117110955

Marley Lobão de Sousa – 120110964

CAMPINA GRANDE/PB, 2021

Sumário

| | | |
|----------|-----------------------------------|-----------|
| 1 | Motivação | 1 |
| 2 | Objetivo | 1 |
| 3 | Fundamentação Teórica | 1 |
| 3.1 | Sensor de Temperatura DS18B20 | 1 |
| 3.2 | Placas de Peltier | 3 |
| 3.3 | Arduino | 4 |
| 3.4 | Sensor RTC | 5 |
| 3.4.1 | DS1302 | 6 |
| 3.4.2 | DS1307 | 7 |
| 3.4.3 | DS3231 | 9 |
| 3.4.4 | Diferenças entre os Módulos | 10 |
| 3.5 | <i>LED</i> | 10 |
| 3.6 | <i>Buzzer</i> | 11 |
| 3.7 | Relé | 12 |
| 3.8 | <i>Qt Creator</i> | 13 |
| 3.9 | <i>C++</i> | 13 |
| 4 | Sistema Físico | 13 |
| 5 | Interface | 14 |
| 5.1 | Temperatura | 15 |
| 5.2 | Data e Horário | 15 |
| 5.3 | Alimentação dos Peixes | 16 |
| 5.4 | Limpeza do Aquário | 16 |
| 6 | Considerações Finais | 17 |
| 7 | Referências Bibliográficas | 17 |

1 Motivação

Atualmente, devido a diversas pesquisas, sabe-se que os aquários servem não apenas para embelezar e decorar o ambiente, mas também para reduzir a ansiedade e o estresse [1], seja cuidando, seja observando. Dessa forma, evidencia-se o motivo do aumento do aquarismo - atividade focada em criar peixes, plantas e organismos aquáticos em conjunto com o senso estético.

Contudo, visto que o aquarismo requer muita atenção e cuidado constante, ele se mostra uma atividade sensível e delicada. Logo, visando ao desenvolvimento saudável e à manutenção de um aquário, diversas variáveis devem ser levadas em consideração, como a temperatura ideal para a água, a iluminação, os horários de alimentação dos peixes, as datas de higienização e de troca de água, entre outras.

Sendo assim, tendo acesso a esses dados, faz-se possível auxiliar no aquarismo a partir da observação constante das variáveis do ambiente e da comunicação do que é necessário fazer de acordo com a situação observada.

2 Objetivo

Este documento tem por objetivo apresentar a execução do sistema físico e da interface do projeto de Aquisição de Dados de um Aquário, para a disciplina de Sistema de Aquisição de Dados e Interface.

3 Fundamentação Teórica

Esta seção busca explicar a teoria dos componentes e apresentar os *softwares* utilizados para a realização do projeto.

3.1 Sensor de Temperatura DS18B20

O sensor de temperatura é um instrumento que ajuda a identificar a variação da temperatura, aquecimento de um ambiente, ou superfície de um material. O sensor utilizado neste projeto é o DS18B20, Figura 1, que está embutido em uma sonda de medição e envolto por uma bainha de aço inoxidável. Cada pino do chip é separado por um tubo termo-encolhível para evitar curto-circuito, permitindo uma vedação interna, e fazendo com que a mesma seja à prova d'água e à prova de umidade. Portanto, essa sonda é adequada para medição de temperatura em ambientes com condições adversas, tal como em um aquário.

3.2 Placas de Peltier

Segundo BENTO [5], Jean Charles Athanase Peltier descobriu, em 1834, que a passagem de uma corrente elétrica através da interface formada por dois condutores diferentes gera um gradiente de temperatura na junção de semicondutores. Assim, dependendo do sentido da corrente elétrica é possível aquecer ou resfriar as placas.

Naturalmente, o calor sempre flui de um corpo mais quente para o corpo mais frio. Portanto, para fazer com que um dos lados fique mais frio é necessário realizar a passagem de corrente por uma ou mais junções de materiais semicondutores do tipo-n e do tipo-p. Uma pastilha de Peltier contém vários elementos semicondutores, agrupados como pares, como na Figura 3.

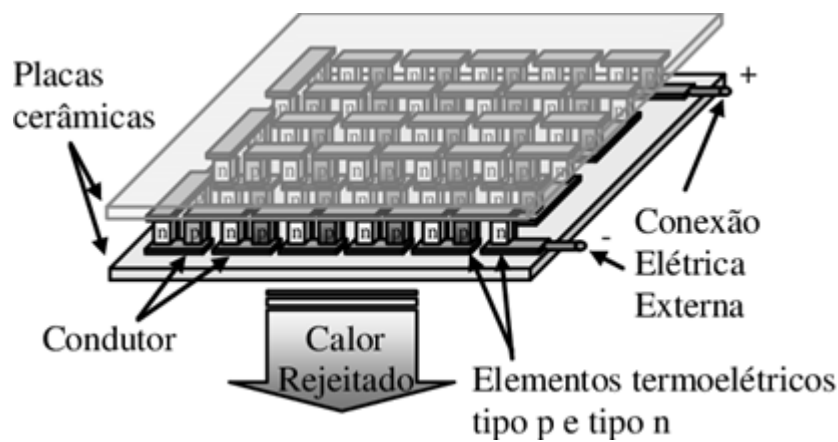


Figura 3 – Desenho esquemático das placas de Peltier

Fonte – JUNIOR ET AL [6]

O princípio de funcionamento se dá ao aplicar-se uma tensão direta à junção p-n, pois pares de elétron/lacuna são criadas perto da junção. Assim, elétrons migram para regiões de maior potencial elétrico, em movimento contrário às lacunas no material tipo p, absorvendo o calor do processo.

O resfriamento ocorre quando elétrons passam do nível baixo de energia no semicondutor do material tipo-p para um nível de energia mais elevado no material semicondutor tipo-n. O calor absorvido é então transferido para a próxima junção, que se encontra na temperatura quente, sendo liberado à medida que os elétrons retornam ao nível baixo de energia no material tipo-p. Esse conjunto de elementos é fixado entre duas placas cerâmicas, eletricamente em série e termicamente em paralelo, fornecendo calor de forma homogênea às superfícies, permitindo um fluxo térmico constante por unidade de área. Assim, efetivamente bombeia-se calor da fonte fria para a superfície quente [7].

Na Figura 4, podemos visualizar o módulo Peltier com o objetivo de refrigerar água de um aquário. Além disso, é possível citar sistemas de refrigeração para sistemas eletrônicos, processadores, bebedouros, aquários, etc.



Figura 4 – Sistema de refrigeração de um aquário

Fonte – AQUARIUM STUFFS [8]

3.3 Arduino

Arduino é uma plataforma de prototipagem eletrônica de *hardware* livre e de placa única, projetada com um microcontrolador Atmel AVR com suporte de entrada/saída embutido, uma linguagem de programação padrão, a qual tem origem em *Wiring*, e é essencialmente *C/C++*. O objetivo do projeto é criar ferramentas que são acessíveis, com baixo custo, flexíveis e fáceis de se usar por principiantes e profissionais. Principalmente para aqueles que não teriam alcance aos controladores mais sofisticados e ferramentas mais complicadas [9].

Pode ser usado para o desenvolvimento de objetos interativos independentes, ou ainda para ser conectado a um computador hospedeiro. Uma típica placa Arduino é composta por um controlador, algumas linhas de E/S digital e analógica, além de uma interface serial ou USB, para interligar-se ao hospedeiro, que é usado para programá-la e interagi-la em tempo real [9].

A grande vantagem em utilizar o arduino em vez do microcontrolador puro é a sua IDE (Ambiente de desenvolvimento integrado), no qual é munida de bibliotecas que facilitam o acesso e utilização dos registradores do microcontrolador, tornando as aplicações mais fáceis, além do fato de possuir uma grande comunidade de desenvolvedores independentes, que compartilham seus projetos e suas bibliotecas.

O arduino escolhido para ser utilizado no projeto foi o Arduino Mega, Figura 5, que possui as seguintes especificações:

- Microcontrolador: ATmega2560;
- Tensão de Operação: 5 V;
- Tensão de Entrada: 7 - 12 V;
- Portas Digitais: 54 (15 podem ser usadas como PWM);
- Portas Analógicas: 16;
- Corrente Pinos I/O: 40 mA;
- Corrente Pinos 3.3 V: 50 mA;
- Memória Flash: 256 KB (8 KB usado no *bootloader*);
- SRAM: 8 KB;
- EEPROM: 4 KB;
- Velocidade do *Clock*: 16 MHz;



Figura 5 – Arduino Mega

Fonte – WIKIPEDIA [9]

3.4 Sensor RTC

O RTC Arduino ou *Real Time Clock* é um módulo eletrônico que mantém o controle da hora atual e pode ser usado em qualquer dispositivo que precise manter a hora precisa

(GPS, registradores de dados, etc), sendo desenvolvido a partir de uma pequena placa, na forma de circuito integrado, com alguns componentes, oscilador e uma bateria embutida.

Os RTCs geralmente têm uma fonte alternativa de energia, para que possam continuar a marcar o tempo enquanto a fonte primária de energia está desligada ou indisponível. Eles costumam usar um oscilador de cristal, que deve ser pequeno, com largura adequada e consumo reduzido de energia, de 32,768 kHz, podendo trabalhar com microcontroladores, como o Arduino, por exemplo.

As vantagens do uso de um RTC incluem um baixo consumo de energia, a liberação do tempo do sistema a partir do cálculo do tempo, a alta precisão e o funcionamento mesmo com falta de energia.

Os três modelos mais usados com Arduino são:

- RTC DS1302;
- RTC DS1307;
- RTC DS3231.

Todos possuem preços que não ultrapassam os 20 reais e podem ser facilmente incorporados em projetos eletrônicos.

3.4.1 DS1302

O RTC 1302 é o mais simples dos três. É capaz de fornecer informações de segundo, minutos, dia, data, mês e ano utilizando um mecanismo de relógio interno com 31 *bytes* de RAM estática. Meses com menos de 31 dias e anos bissextos são tratados automaticamente pelo módulo. Opera nos formato 12, com indicador AM e PM, e 24 horas.

O módulo, Figura 6, vem com uma bateria de longa duração, cerca de 5 anos, de íon de lítio de 3V, um cristal de 32 kHz e um chip DS1302 em uma pequena placa com cinco pinos.

Esse módulo opera com tensão de alimentação de 5V e usa o protocolo serial I2C para comunicação, o que facilita muito a interface dele com o microcontrolador. Apenas dois pinos (CLK e DAT) são necessários para fazer a interface com o Arduino, exceto VCC e GND, enquanto RST é o pino de reinicialização (opcional).

Apenas três fios são necessários para se comunicar com a memória de acesso aleatório ou RAM do relógio do DS1302. São eles: CE (*Chip Enable*), IO (*Input Output Data Line*) e SCLK (*Serial Clock*). Esse módulo é projetado para operar com energia muito baixa e reter dados e informações do relógio em menos de 1 μ W.

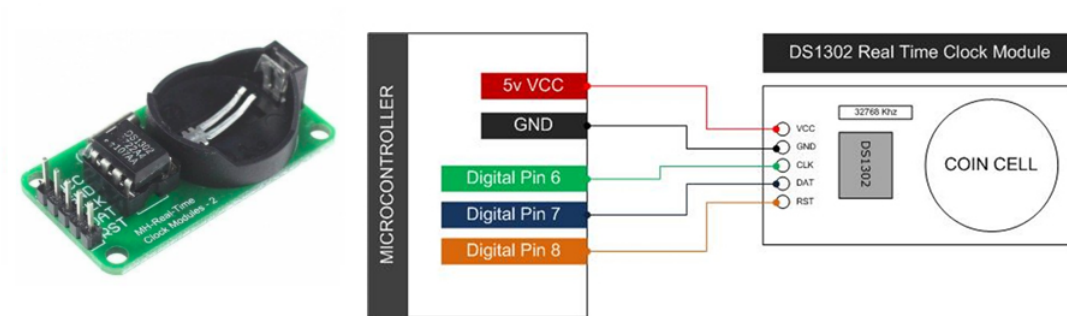


Figura 6 – RTC DS1302

Fonte – Adaptada de USINAINFO [10] e VIDAL [11]

As principais especificações desse módulo são:

- Chip: DS1302;
- 31 *Bytes* de memória RAM para armazenamento de dados;
- Tensão: 2,0V - 5,5V;
- Corrente: menos de 300nA em 2V e até 1,2mA em 5V;
- Sinal de saída programável em onda quadrada;
- Consome menos de 300nA no modo oscilador.

3.4.2 DS1307

O RTC DS1307, Figura 7, é um dos módulos RTCs mais acessíveis e comuns. Esse módulo é um relógio de tempo real com calendário completo e mais de 56 *bytes* de SRAM de memória não-volátil. Tal como os outros RTCs, fornece informações de segundo, minutos, dia, data, mês e ano. Em sua placa há um circuito que detecta falhas de energia e aciona a bateria embutida com objetivo de evitar perda de dados. Ele Utiliza protocolo I2C e trata automaticamente os casos de meses com menos de 31 dias e anos bissextos.

Algumas de suas vantagens são:

- Capacidade de gerar ondas quadradas programáveis;
- Uso de baixa corrente - abaixo de 500nA;
- A capacidade de definir a data até o ano 2100;
- Interface I2C Serial.

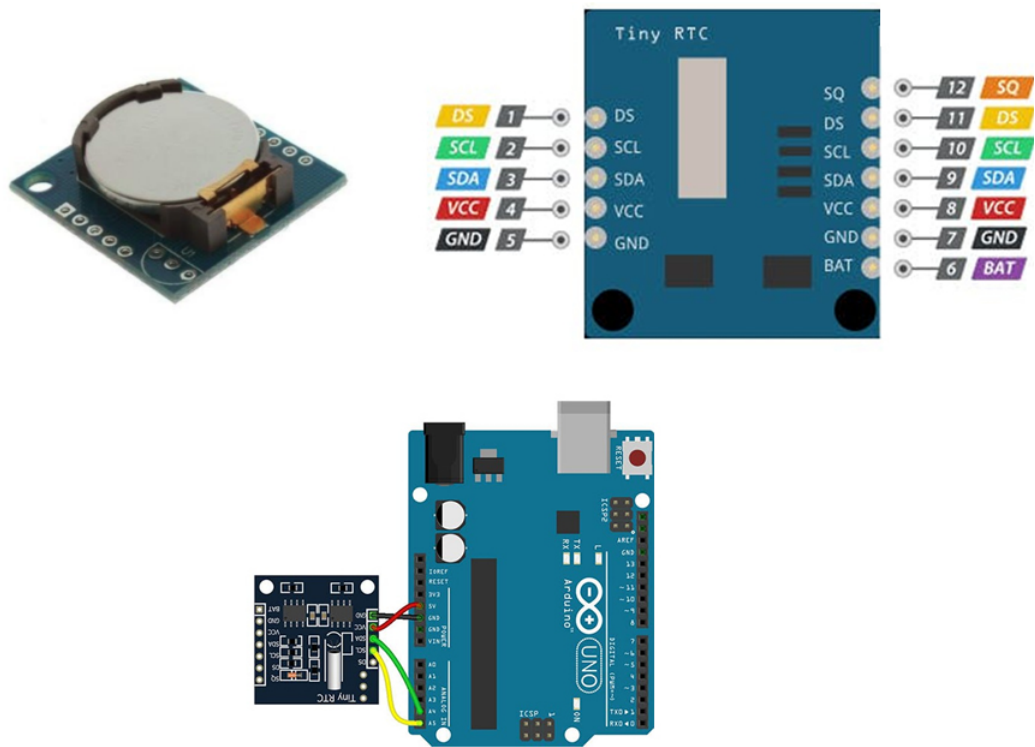


Figura 7 – RTC DS1307

Fonte – Modificada de [10] e [11]

O módulo DS1307 tem a capacidade de instalar uma bateria reserva CR2023 de 3 volts, cuja tensão pode ser lida no pino BAT. Possui também uma memória EEPROM 24c32 embutida que pode salvar 32kb de dados. Além disso, permite a instalação de um sensor DS18B20 integrado para medição da temperatura.

As principais especificações desse módulo são:

- Chip: DS1307;
- Computa segundos, minutos, horas, dias da semana, dias do mês, meses e anos (de 2000 a 2099);
- 56 *bytes* de SRAM;
- Tensão: 4,5 - 5,5V;
- Circuito de detecção de falha de energia;
- Consome menos de 500nA no modo bateria com oscilador em funcionamento;
- Faixa de temperatura: -40°C a +85°C;
- Dimensões: 2,9×2,6cm.

3.4.3 DS3231

O DS3231, Figura 8, é um RTC de alta precisão e baixo consumo de energia. A sua precisão se deve ao uso de um oscilador de cristal com compensação de temperatura (TCXO). Ele funciona tanto no formato 12 horas como 24 horas, e as informações de meses com menos de 30/31 dias e anos bissextos são corrigidos automaticamente pelo módulo.

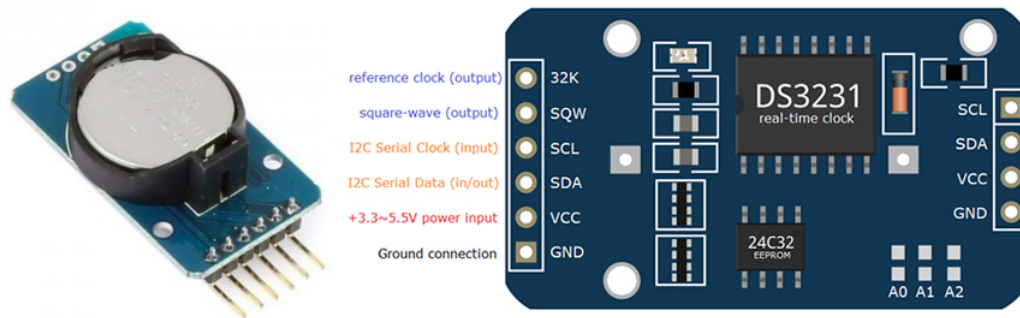


Figura 8 – RTC DS3231

Fonte – Modificada de [10] e [11]

Uma bateria acompanha o módulo para evitar a perda de dados em caso de desenergização do circuito. Um aspecto importante e que diferencia dos outros modelos é o protocolo de comunicação utilizado. No caso do DS3231, o protocolo I2C é usado com um endereço padrão fixo de 0x68. Assim, são dois canais(pinos) de comunicação: SCL(Clock) e SDA(dados).

Além do chip RTC, este módulo em particular também possui um chip EEPROM 24C32. Um EEPROM é um tipo de dispositivo de armazenamento de dados em que você pode ler / gravar dados. Ele compartilha o barramento I2C do módulo com o DS3231 e tem o endereço padrão de 0x57.

Os pinos e suas respectivas funcionalidades são:

- 32K - oscilador de 32KHz muito preciso;
- SQW - emite um sinal de onda quadrada (1Hz, 4kHz, 8kHz ou 32kHz) e pode ser usado como uma saída de interrupção;
- SCL - pino de entrada para relógio com comunicação serial I2C;
- SDA - pino de entrada/saída para dados com comunicação serial I2C;
- VCC - pino de entrada da fonte de alimentação para o módulo;
- GND - Conexão do pino de aterramento;

Os pinos SCL , SDA , VCC e GND no lado direito do módulo são conectados internamente nos pinos do lado esquerdo com a mesma etiqueta.

As principais especificações desse módulo são:

- Tensão de operação: 3,3 - 5V;
- Computa segundos, minutos, horas, dias da semana, dias do mês, meses e anos (de 2000 a 2099);
- Sensor de temperatura com 3°C de exatidão;
- Chip de memória: AT24C32 de 32kb que pode ser usado como RAM estendida do microcontrolador;
- Circuito de detecção de falha de energia;
- Consumo menor que 500nA no modo bateria com oscilador em funcionamento;
- Dimensões: 38 x 22 x 14mm;
- Peso: 8g.

3.4.4 Diferenças entre os Módulos

As principais diferenças entre os módulos dizem respeito ao protocolo utilizado, tamanho e tipo da memória interna, dimensões e consumo de energia. Os módulos DS1307 e DS3231 utilizam protocolo I2C, enquanto o DS1302 utiliza um protocolo serial. O protocolo I2C utiliza 2 pinos: o SDA(dados) e o SCL(*clock* de sincronismo). Já o DS1302 utiliza três pinos: SCL(*clock* de sincronismo), CE(*Chip Enable* – para habilitar a transmissão de dados) e I/O(dados) [11].

As tensões de alimentação também são diferenças importantes, com o DS1302 operando com um mínimo de 2V, enquanto o DS1307 opera com no mínimo 4,5V [11].

3.5 LED

O LED, *Light-emitting Diode*, é um dispositivo semicondutor utilizado para emissão de luz. Tal transformação de energia elétrica em luz é diferente da encontrada nas lâmpadas convencionais que utilizam filamentos metálicos, radiação ultravioleta e descarga de gases, dentre outras. Nos LEDs, a transformação de energia elétrica em luz é feita na matéria, sendo, por isso, chamada de Estado Sólido (*Solid State*)[12].

O LED, Figura 9, é um componente do tipo bipolar, ou seja, tem um terminal chamado anodo e outro, chamado catodo. Dependendo de como for polarizado, permite ou não a passagem de corrente elétrica e, conseqüentemente, a geração ou não de luz.

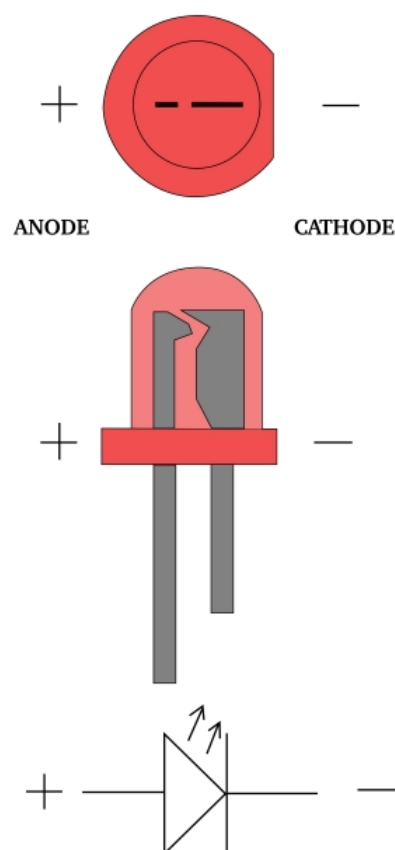


Figura 9 – LED

Fonte – WIKIPEDIA [13]

3.6 Buzzer

O buzzer ativo, Figura 10, é um dispositivo que possui um oscilador interno que, ao ser energizado, consegue emitir um sinal sonoro que é muito utilizado por projetistas de dispositivos como forma de alerta. O buzzer utilizado possui as seguintes especificações:

- Tensão de operação: 4-8 V_{DC} ;
- Corrente de operação: 30 mA;
- Saída de som mínima (a 10 cm): 85 dB;
- Frequência de ressonância: 2300 Hz com margem de 300 Hz;
- Temperatura de operação: -27 a +70 °C



Figura 10 – Buzzer

Fonte – USINAINFO [14]

3.7 Relé

O relé, Figura 11, é um componente eletromecânico, que aciona um interruptor a partir de um sinal. Esse interruptor possui elevada capacidade de tensão e corrente. Eles são amplamente utilizados em automação residencial e comercial.

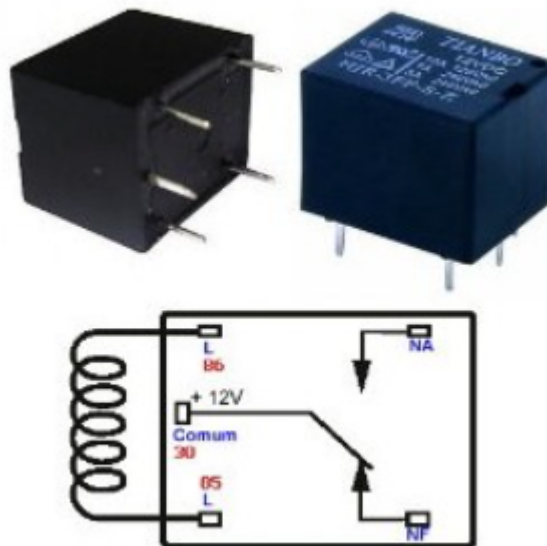


Figura 11 – Relé

Fonte – ATHOS ELETRONICS [15]

O relé possui uma bobina, e um contato preso a uma mola de rearme, responsável por conectar os terminais nas posições Normalmente Aberto (NA) e Normalmente Fechado (NF). No repouso, o contato está na posição NF. Já na energização da bobina, cria-se um

campo eletromagnético que atrai e desloca o contato. Isso acontece sem misturar os sinais, já que a bobina é totalmente isolada dos contatos que serão chaveados.

3.8 *Qt Creator*

O *Qt Creator* é um ambiente de desenvolvimento integrado *C++*, *JavaScript* e *QML* de plataforma cruzada que simplifica o desenvolvimento de aplicativos GUI (*Graphical User Interface*). O desenvolvimento do que eventualmente se tornaria o *Qt Creator* começou em 2007 e o lançamento de sua versão 1.0 ocorreu em março de 2009 [16].

3.9 *C++*

C++ é uma linguagem de programação compilada multiparadigma, incluindo uma linguagem imperativa, orientada a objetos e de uso geral. Desde os anos 1990 é uma das linguagens comerciais mais populares, sendo bastante usada também na academia por seu grande desempenho e base de utilizadores [17].

Bjarne Stroustrup desenvolveu o *C++* em 1983 no Bell Labs como um adicional à linguagem *C*. Novas características foram adicionadas com o tempo, como funções virtuais, sobrecarga de operadores, herança múltipla, gabaritos e tratamento de exceções. Após a padronização ISO realizada em 1998 e a posterior revisão realizada em 2003, uma nova versão da especificação da linguagem foi lançada em dezembro de 2014 [17].

4 Sistema Físico

O sistema físico da aquisição de dados de um aquário se encontra na Figura 12.

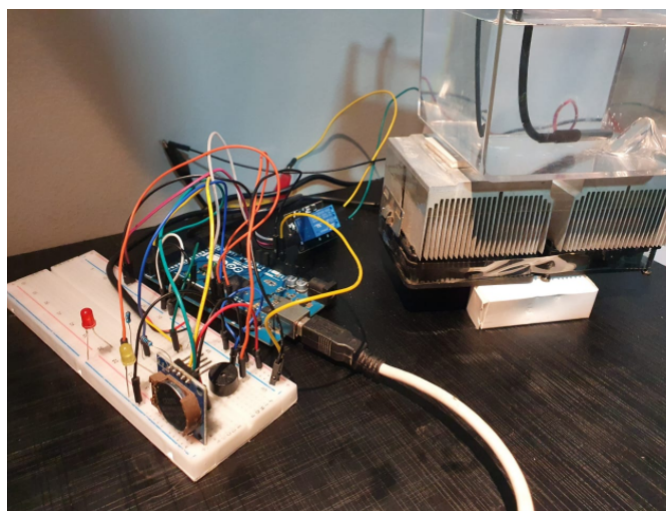


Figura 12 – Sistema de Aquisição de Dados de um Aquário

Fonte – Autoria Própria (2021)

Nesse circuito, o sensor é acoplado ao arduino de forma a, utilizando a biblioteca do dispositivo DS18B20, ler o valor da temperatura do ambiente, no caso, será lido o valor da temperatura da água do aquário. Em conjunto, se apresentam o *led* vermelho para indicar temperatura fora dos níveis pré-definidos, o *led* amarelo para indicar que o aquário precisa ser limpo e o *buzzer* para indicar que os peixes precisam ser alimentados.

Concomitantemente, tem-se o circuito para a implementação do RTC, que consegue obter a data e o horário com alta precisão e um baixo consumo de energia. Seus valores também são enviados pela porta serial do arduino.

Por fim, implementou-se as placas de Peltier, que controlam a temperatura do aquário, de forma a manter a mesma dentro dos limites estabelecidos pelo usuário.

A implementação do sistema físico com a interface pode ser visualizada na Figura 13.

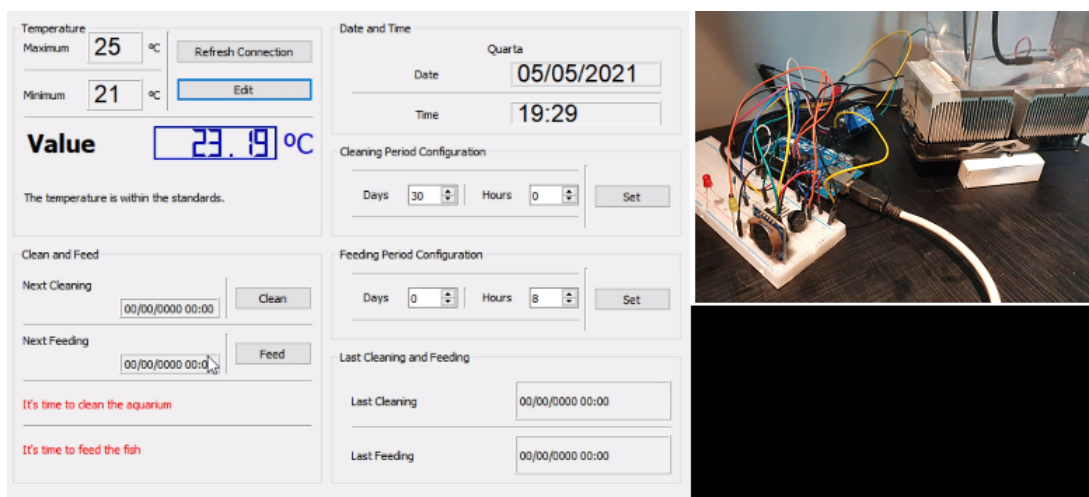


Figura 13 – Integração do Sistema Físico de Aquisição de Dados de um Aquário com sua Interface

Fonte – Autoria Própria (2021)

5 Interface

A interface foi implementada utilizando o *Qt Creator* na linguagem *C++*. Dessa forma, torna-se possível exibir os valores adquiridos e as mensagens de atenção ou situação do aquário de forma simples e direta a partir da comunicação serial bidirecional com o Arduino. A interface da Aquisição de Dados de um Aquário pode ser observada na Figura 14.

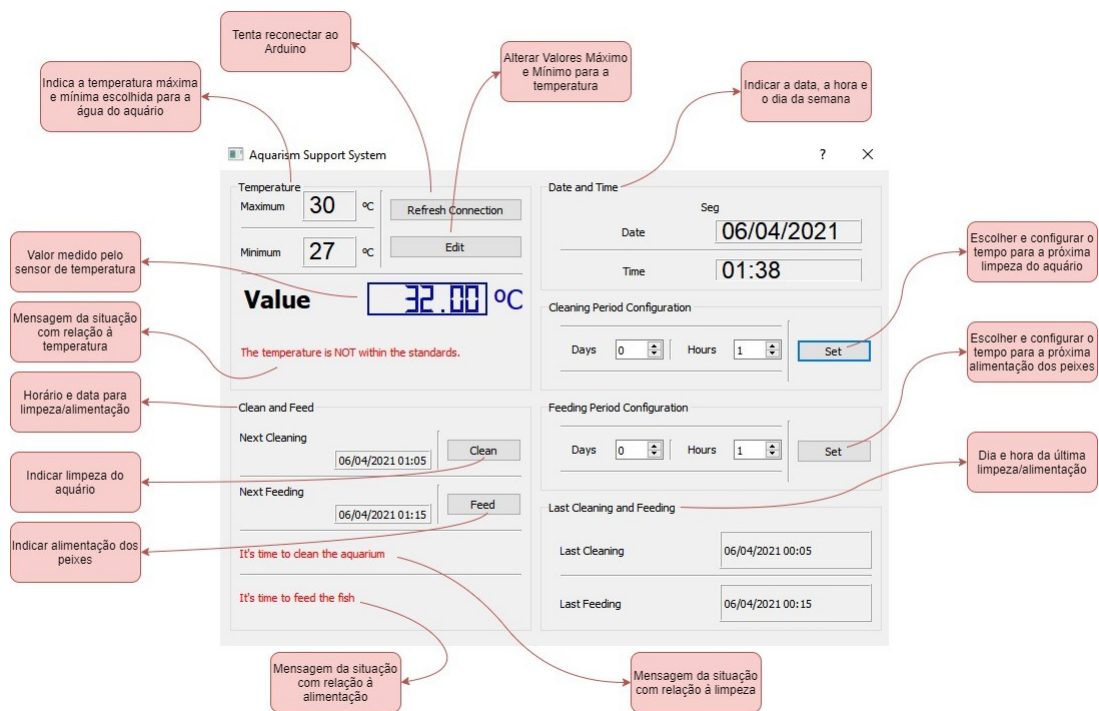


Figura 14 – Interface do Sistema de Aquisição de Dados de um Aquário

Fonte – Autoria Própria (2021)

5.1 Temperatura

O valor da temperatura é capturado utilizando o sensor DS18B20 em conjunto com o Arduino. A leitura é enviada para a interface via comunicação serial e exibida no bloco *Temperature*. No mesmo bloco, os limites máximo (30°C) e mínimo (27°C) de temperatura são indicados juntamente com os botões *Refresh Connection*, para reconectar ao Arduino e o botão *Edit*, para alterar os valores máximo e mínimo. Caso a temperatura lida esteja fora do intervalo indicado, a mensagem *The temperature is NOT within the standards* é exibida, indicando a necessidade de corrigir a temperatura da água para melhor acomodação dos peixes.

Quando isso ocorre, a interface manda uma mensagem para o arduino indicando se a temperatura está abaixo da mínima ou acima da máxima para que seja possível implementar um controle da temperatura e acionar o indicador do alarme.

5.2 Data e Horário

O horário e a data são obtidos utilizando o sensor RTC (*Real Time Clock*), que é um relógio de tempo real de alta precisão e baixo consumo de energia. Os valores são enviados pela comunicação serial de forma a exibir, no bloco *Date and Time*, os valores da data, do horário e do dia da semana obtidos a partir do sensor.

Esses valores são utilizados para registrar os momentos em que a alimentação e

limpeza são realizados, para calcular os momentos da próxima alimentação e da próxima limpeza e para comparar se o momento atual está antes ou depois do calculado.

Dessa forma, quando passa do momento de alimentar os peixes ou de limpar o aquário, uma mensagem é enviada para o controlador via comunicação serial a fim de ativar o alarme que lembre o usuário de efetuar a tarefa.

Uma possível melhoria é a implementação de um sistema automático de alimentação e de limpeza para serem executados no período programado.

5.3 Alimentação dos Peixes

Nesta etapa, com o uso da sincronização do horário feita pelo sensor RTC, o usuário pode verificar o horário para alimentar os peixes de acordo com a configuração do período de alimentação. Essa configuração pode ser modificada pelo usuário caso o mesmo ache que o valor *default* (8 horas) seja impróprio. Para modificar essa configuração, o usuário deve acessar o bloco *Feeding Period Configuration* e selecionar o tempo entre as refeições, uma vez selecionado, deve-se clicar no botão *Set* para enviar essa informação para o sistema, que imediatamente atualizará o bloco *Next Feeding* com a data e o horário da próxima refeição.

Caso passe do horário de alimentar os peixes e os mesmos não tenham sido alimentados, um *buzzer* e um *led* são acionados e a mensagem *It's time to feed the fish* é mostrada na área de mensagens da interface para alertar ao usuário que está na hora de alimentar os peixes. Esse *buzzer* e esse *led* param o seu funcionamento quando o usuário aperta o botão *feed*, então a interface atualiza a mensagem para *fish fed*, atualiza a data e o horário da última refeição no bloco *Last Feeding* e atualiza a data e o horário para a próxima refeição no bloco *Next Feeding*.

5.4 Limpeza do Aquário

De forma semelhante à alimentação dos peixes, a limpeza do aquário pode ter o período configurado no bloco *Cleaning Period Configuration* ao selecionar o tempo entre as limpezas e apertar no botão *Set*, atualizando também a data e o horário da próxima limpeza do aquário no bloco *Next Cleaning*.

O sistema de alarme com um *buzzer*, um *led* e a mensagem *It's time to clean the aquarium* também é utilizado para chamar atenção do usuário quando o horário de limpeza é ultrapassado. O alarme é interrompido ao apertar o botão *Clean*, que também atualiza as datas e os horários da última limpeza e da próxima limpeza, respectivamente nos blocos *Last Cleaning* e *Next Cleaning*.

6 Considerações Finais

A solução proposta exige uma rotina de programação que seja capaz de realizar ações e alertas de acordo com as condições apresentadas pelas medidas mensuradas pelos sensores. Foi então desenvolvido um código no arduino e no *Qt Creator* que pudesse sanar essas necessidades. Em que os testes de integração do sistema físico com a interface foram realizados com sucesso.

Os códigos utilizados para a configuração do arduino e da interface estão disponibilizados no [github](#). Inicialmente o usuário deve carregar o arquivo *Arduino/AquarismSupportSystem/ArduinoAquarism.ino* no arduino, em sequência deve-se baixar o zip com o executável *AquarismSupportSystemExe*, extrair e abrir o executável *AquarismSupportSystem.exe* e pronto, o sistema estará configurado.

O sistema físico foi capaz de ler a temperatura por meio do sensor DS18B20 acoplado ao arduino e passar essas informações para a interface pela comunicação serial. Além disso, o mesmo alerta por meio dos *leds* vermelho, amarelo e do *buzzer* quando a temperatura está fora dos limites estabelecidos, ou se está na hora de limpar o aquário ou de alimentar os peixes, respectivamente. Além disso, as placas de Peltier ajustam a temperatura do sistema quando a mesma ultrapassa os limites estabelecidos.

A interface realizada pode ser acessada via arquivo executável, sem a necessidade de se instalar o *Qt Creator*, e atende ao objetivo proposto: alertar ao usuário caso a temperatura esteja elevada ou muito baixa; alertar caso os peixes não tenham sido alimentados no horário indicado; alertar caso o aquário não tenha sido limpo no tempo previsto; informar as datas e horários das próximas limpeza e alimentação, bem como mostrar as datas e horários das últimas limpeza e alimentação.

7 Referências Bibliográficas

- [1] PETABLE. Peixes de aquário ajudam a reduzir stress. Disponível em: <<https://petable.care/pt/2017/11/15/15-formas-como-os-peixes-de-aquario-ajudam-reduzir-o-stress/>>. Acesso em: 13 Abr. 2021.
- [2] Aquários e Peixes Ornamentais. Como cuidar do seu aquário – Para pessoas iniciantes. Disponível em: <<http://aquarioepeixes.com.br/peixesornamentais/aquario-de-agua-doce/como-cuidar-do-seu-aquario/>>. Acesso em: 13 Abr. 2021.
- [3] GAIMC. DS18B20 Temperature Sensor. Disponível em: <<https://www.gaimc.com/Uploads/Download/Temperature/GTS200.pdf>>. Acesso em: 29 Abr. 2021.
- [4] MAXIM INTEGRATED. Datasheet do Sensor DS18B20. Disponível em: <<https://storage.googleapis.com/baudaeletronicadasheet/DS18B20.pdf>>. Acesso em: 29 Abr.

2021.

[5] BENTO, J. V. et Al. (2018). Sistema de Refrigeração Termoelétrica de Peltier Usado para Arrefecer o Processador de Computador. Trabalho de Conclusão de Curso. Disponível em: <http://repositorio.aee.edu.br/bitstream/aee/478/1/1_Gruppo_1_Cassio_Felipe_Ricardo.pdf>. Acesso em: 29 Abr. 2021.

[6] JUNIOR, J.; RIBEIRO, E.; SANTOS, D. (2012). Estabilização da Temperatura da Câmara Digital Magi Dc575 utilizando Pastilha Termoelétrica para Obtenção dos Parâmetros de Correção do Efeito da Corrente Escura. Revista Brasileira de Cartografia. Disponível em: <https://www.researchgate.net/publication/277064277_ESTABILIZACAO_DA_TEMPERATURA_DA_CAMARA_DIGITAL_Magi_DC575_UTILIZANDO_PASTILHA_TERMOELETRICA_PARA_OBTENCAO_DOS_PARAMETROS_DE_CORRECAO_DO_EFEITO_DA_CORRENTE_ESCURA>. Acesso em: 29 Abr. 2021.

[7] MOURA, P.; ALMEIDA, D. (2014). Refrigerador termoelétrico de peltier usado para estabilizar um feixe laser em experimentos didáticos. Revista Brasileira de Ensino de Física. 36. 1-5.10.1590/S1806-11172014000100008. Disponível em: <https://www.researchgate.net/publication/274937828_Refrigerador_termoeletrico_de_peltier_usado_para_estabilisar_um_feixe_laser_em_experimentos_didaticos>. Acesso em: 29 Abr. 2021.

[8] Aquarium Stuffs (2011). Chiller com Efeito Peltier funcionado em um aquário marinho doméstico. Disponível em: <<http://aquariumstuffs.blogspot.com/2011/10/>>. Acesso em: 29 Abr. 2021.

[9] WIKIPEDIA. Arduino. Disponível em: <<https://pt.wikipedia.org/wiki/Arduino>>. Acesso em: 13 Abr. 2021.

[10] USINAINFO: Eletrônica & Robótica. RTC ARDUINO. Disponível em: <<https://www.usinainfo.com.br/rtc-arduino-539#>>. Acesso em: 17 Abr. 2021.

[11] VIDAL, V. RTC – Real Time Clock DS1302, 1307 e 3231. BLOG ELETROGATE. Disponível em: <<https://blog.eletrogate.com/rtc-real-time-clock-ds1302-1307-e-3231/>>. Acesso em: 17 Abr. 2021.

[12] LABORATORIO DE ILUMINAÇÃO. LED - o que é e como funciona. Disponível em: <<https://hosting.iar.unicamp.br/lab/luz/dicasemail/led/dica36.htm>>. Acesso em: 05 Mai. 2021.

[13] WIKIPEDIA. Diodo emissor de luz. Disponível em: <https://pt.wikipedia.org/wiki/Diodo_emissor_de_luz>. Acesso em: 02 Mai. 2021.

[14] USINAINFO. Buzzer arduino ativo e passivo. Disponível em: <<https://www.usinainfo.com.br/buzzer-arduino-535>>. Acesso em: 05 Mai. 2021.

[15] ATHOS ELETRONICS. Relé - o que é e como funciona. Disponível em: <<https://athoselectronics.com/rele/>>. Acesso em: 05 Mai. 2021.

[16] QT CREATOR. Qt Creator - A Cross-platform IDE for Application Development. Disponível em: <<https://www.qt.io/product/development-tools>>. Acesso em: 13 Abr. 2021.

[17] CPLUSPLUS.COM. A Brief Description. Disponível em: <<https://www.cplusplus.com/info/description/>>. Acesso em: 13 Abr. 2021.