UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO

RODRIGO REZENDE SIMÕES DA SILVA VIEIRA

INSTRUMENTAÇÃO E CONTROLE DE UMA BANCADA DIDÁTICA DE REFRIGERAÇÃO

VITÓRIA
2017

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO

RODRIGO REZENDE SIMÕES DA SILVA VIEIRA

INSTRUMENTAÇÃO E CONTROLE DE UMA BANCADA DIDÁTICA DE REFRIGERAÇÃO

Parte manuscrita do Projeto de Graduação do aluno Rodrigo Rezende Simões da Silva Vieira, apresentada ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito para obtenção do grau de Bacharel em Engenharia Elétrica.

ORIENTADOR: PROF. DR. HANS JORG ANDREAS SCHNEEBELI

VITÓRIA
2017

INSTRUMENTAÇÃO E CONTROLE DE UMA BANCADA DIDÁTICA DE REFRIGERAÇÃO

RODRIGO REZENDE SIMÕES DA SILVA VIEIRA

BANCA EXAMINADORA

Prof. Dr. Hans Jorg Andreas Schneebeli
Universidade Federal do Espírito Santo
Orientador

Prof. Dra. Eliete Maria de Oliveira Caldeira
Universidade Federal do Espírito Santo

Prof. Dr. Alessandro Mattedi
Universidade Federal do Espírito Santo

VITÓRIA
2017

RESUMO

Este trabalho propõe o projeto de instrumentação e controle de uma bancada didática de refrigeração. Foi utilizado um microcontrolador com tecnologia ARM e sensores de umidade, pressão e temperatura. Os dados coletados foram mostrados em uma interface gráfica no computador, que se comunica via USB com o microcontrolador, utilizando um protocolo próprio de dados. Foi utilizado o FreeRTOS como ferramenta na elaboração do software, sendo projetado um sistema de tempo real.

Palavras-chave: Automação. Sistemas Embarcados. Eletrônica. Bancada de Refrigeração. FreeRTOS

LISTA DE FIGURAS

Figura 1 – Desenho da bancada de refrigeração didática 8
Figura 2 – Ciclo térmico da bancada em modos refrigeração ou ar condicionado..... 9
Figura 3 – Diagama de blocos do sistema 13
Figura 4 – Sensores de temperatura (Ds18b20), umidade (DHT11) e pressão (SB69 500V) e microcontrolador (STM32f103c8t6) utilizados no projeto..... 15
Figura 5 – Equemático dos periféricos utilizados..... 17
Figura 6 – Ligações no microcontrolador..... 18
Figura 7 – Circuito de acionamento das cargas 19
Figura 8 – Circuito de comunicação com os sensores de umidade e temperatura 20
Figura 9 – Diagrama simplificado das rotinas do sistema..... 22
Figura 10 – Estados da função lerSensores()..... 23
Figura 11 – Estrutura do pacote de dados..... 24
Figura 12 – Dados dos sensores mostrados na interface gráfica 27

LISTA DE TABELAS

Tabela 1 – Características das cargas do sistema. 19
Tabela 2 – Códigos ROM dos 11 sensores de temperatura do sistema 21
Tabela 3 – Tempos aproximados para atualização e leitura dos sensores 25
Tabela 4 – Comparação entre os sensores utilizados no projeto e sensores comerciais..... 26

LISTA DE ABREVIATURAS E SIGLAS

ARM	*Advanced RISC Machine*
GPIO	*General Purpose Input/Output*
PCI	Placa de Circuito Impresso
UFES	Universidade Federal do Espírito Santo
EEPROM	*Electrically Erasable Programmable Read-Only Memory*
ROM	*Read-Only Memory*
USB	*Universal Serial Bus*
I/O	*Input/Output*

SUMÁRIO

1 INTRODUÇÃO 8
1.1 Motivação 8
1.2 Definição do Problema 9
1.3 Objetivos Específicos 10
1.4 Metodologia 11
1.4 Estrutura do Trabalho 11
2 SISTEMA 13
2.1 Arquitetura Simplificada 13
2.2 Arquitetura Detalhada 13
2.3 Arquitetura do *Software* 15
2.3 Arquitetura do *Hardware* 17
3 IMPLEMENTAÇÃO DO *HARDWARE* 18
4 IMPLEMENTAÇÃO DO *SOFTWARE* 21
5 TESTES 25
6 CONCLUSÃO E TRABALHOS FUTUROS 28
REFERÊNCIAS 29
ANEXO A 31
ANEXO B 35

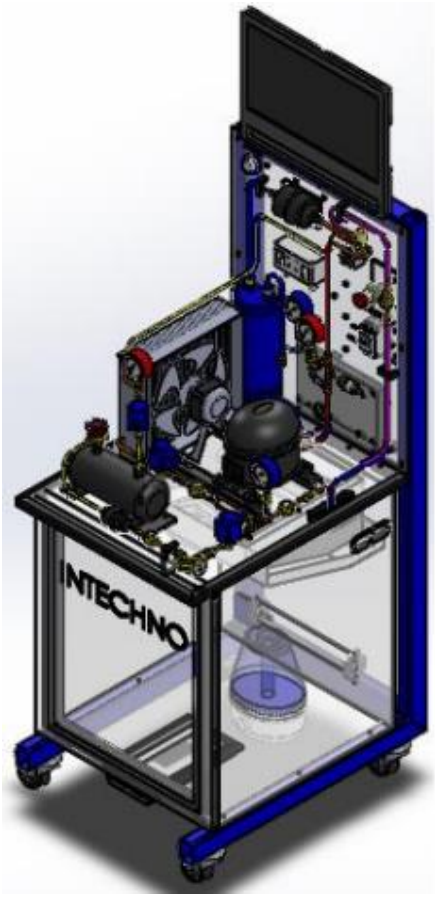
1. INTRODUÇÃO

1.1 Motivação

Os conceitos vistos nas aulas teóricas são colocados à prova durante as aulas práticas, momento em que o aluno se depara com situações reais que envolvem a matéria estudada. Com o objetivo de enriquecer a formação dos estudantes são utilizadas bancadas didáticas, com estruturas físicas semelhantes às encontradas comercialmente, porém com uma gama de sensores, visando o estudo ponto a ponto dos fenômenos envolvidos no sistema.

Em cursos técnicos e superiores de mecânica é usada uma bancada didática de refrigeração, mostrada na Figura 1, desenvolvida pela empresa Intechno Equipamentos Didáticos. O principal objetivo desta bancada é possibilitar aos alunos observarem os fenômenos envolvidos durante um ciclo de refrigeração.

Figura 1- Desenho da bancada de refrigeração didática.

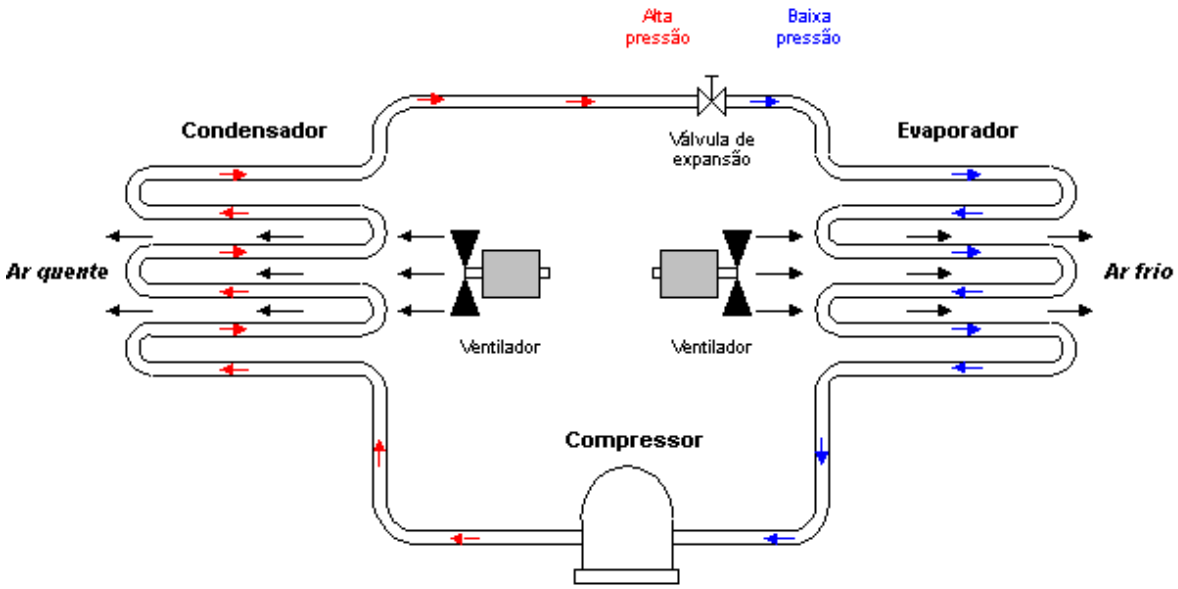


Fonte: INTECHNO, 2017

O ciclo de refrigeração, mostrado na Figura 2, objetiva retirar o calor do ambiente interno e transferi-lo ao ambiente externo. Tudo se inicia no compressor, equipamento responsável pela movimentação do fluido refrigerante no interior das tubulações de cobre. O fluido a alta pressão que sai do compressor entra no condensador, etapa em que perde calor para o ambiente e realiza uma mudança de estado físico, apresentando-se no estado líquido na saída. Já no estado líquido, o mesmo passa por um dispositivo de expansão, responsável por realizar uma queda brusca de pressão e consequentemente de temperatura.

O fluido já a uma temperatura baixa e ainda em estado líquido entra no evaporador, onde realiza a troca térmica com o ambiente no interior da câmara. O fluido aquece e evapora, retornando novamente ao estado gasoso, voltando ao compressor e reiniciando o ciclo.

Figura 2 - Ciclo térmico da bancada em modo refrigeração ou ar condicionado.



Fonte: TERMODINÂMICA APLICADA, 2017

A bancada de refrigeração possibilita então o estudo dos fenômenos descritos acima por meio da utilização de sensores e atuadores, permitindo que o aluno observe as mudanças de pressão e temperatura ao longo do ciclo de refrigeração. O equipamento permite também o controle de alguns equipamentos envolvidos no sistema, possibilitando estudos minuciosos de falhas, o que não seria possível sem um equipamento totalmente instrumentado e automático.

1.2 Definição do Problema

Instrumentar uma bancada de refrigeração didática não instrumentada, desenvolvida pela empresa Intechno Equipamentos Didáticos, mostrando os dados coletados em uma interface computacional desenvolvida pela própria empresa e realizar o controle a partir de entradas dados pelo usuário. O sistema tem como objetivo principal ser de fácil montagem e manutenção, de baixo custo e com componentes encontrados facilmente no mercado brasileiro.

1.3 Objetivos Específicos

Criação de um sistema que seja capaz de ler os dados dos sensores, atuar na planta e se comunicar via USB com um *software* de computador desenvolvido pela própria empresa Intechno, cujo protocolo de comunicação também é de autoria da mesma. Será projetado então o *software* embarcado responsável pela instrumentação, controle e comunicação USB e o *hardware* necessário à atuação dos dispositivos da planta.

Será necessária a coleta dos seguintes dados:

- Temperatura de entrada do condensador;
- Temperatura de saída do condensador;
- Temperatura de entrada do compressor;
- Temperatura de saída do compressor;
- Temperatura de entrada do evaporador;
- Temperatura de saída do evaporador;
- Temperatura do ar do condensador;
- Temperatura do compressor;
- Temperatura do ar do evaporador;
- Temperatura do interior da câmara;
- Temperatura ambiente;
- Umidade relativa ambiente;
- Umidade relativa do interior da câmara;
- Pressão de entrada do compressor;
- Pressão de saída do compressor;

O mesmo também deverá ser capaz de atuar no:

- Acionamento do ventilador do condensador;
- Acionamento do ventilador do evaporador;
- Acionamento da eletroválvula do modo Ar Condicionado;
- Acionamento da eletroválvula do modo Refrigeração;
- Acionamento do compressor;
- Acionamento da válvula reversora;
- Acionamento do umidificador de ar;

1.4 Metodologia

Os sensores utilizados deverão ser capazes de ler temperaturas na faixa de -5 até 50ºC, com precisão de até 10% e taxa de resposta inferior a 300ms. Serão necessários também sensores de umidade capazes de ler umidade relativa do ar de 20 a 90%, com precisão de até 10% e tempo de resposta de até 5s. Já os sensores de pressão deverão ser capazes de medir na faixa de 0 a 250psi com precisão de 10% e taxa de resposta inferior a 300ms.

Será necessário para este projeto um microcontrolador que seja capaz de realizar leitura analógica, possua *timers* para temporização, comunicação USB, número de portaS I/O suficientes e que seja de baixo custo e fácil acesso.

O *software* desenvolvido para o microcontrolador deverá obedecer a uma arquitetura que o faça ser capaz de ler todos os sensores e atuar na planta seguindo instruções vindas de uma interface no computador por meio de comunicação USB, a cada 300ms.

Primeiramente serão testados os sensores e atuadores individualmente, observando a resposta e efetuando as devidas alterações no *software* e *hardware* desenvolvidos. Sanados os possíveis problemas, será realizada a montagem completa e a observação dos tempos de resposta dos sensores na interface, verificando se foram obedecidos os tempos de atualização previamente definidos. Para o teste final será realizado o controle de temperatura do interior da câmara.

1.5 Estrutura do Trabalho

No Capítulo 2 será apresentado o sistema utilizado, discutindo as escolhas dos sensores, da comunicação e da arquitetura empregada.

No Capítulo 3 será descrito o projeto do *hardware*, apresentando os circuitos utilizados.

No Capítulo 4 será descrita a implementação do *software* e as ferramentas utilizadas no seu desenvolvimento.

No Capítulo 5 serão mostrados os testes realizados e a validação do sistema.

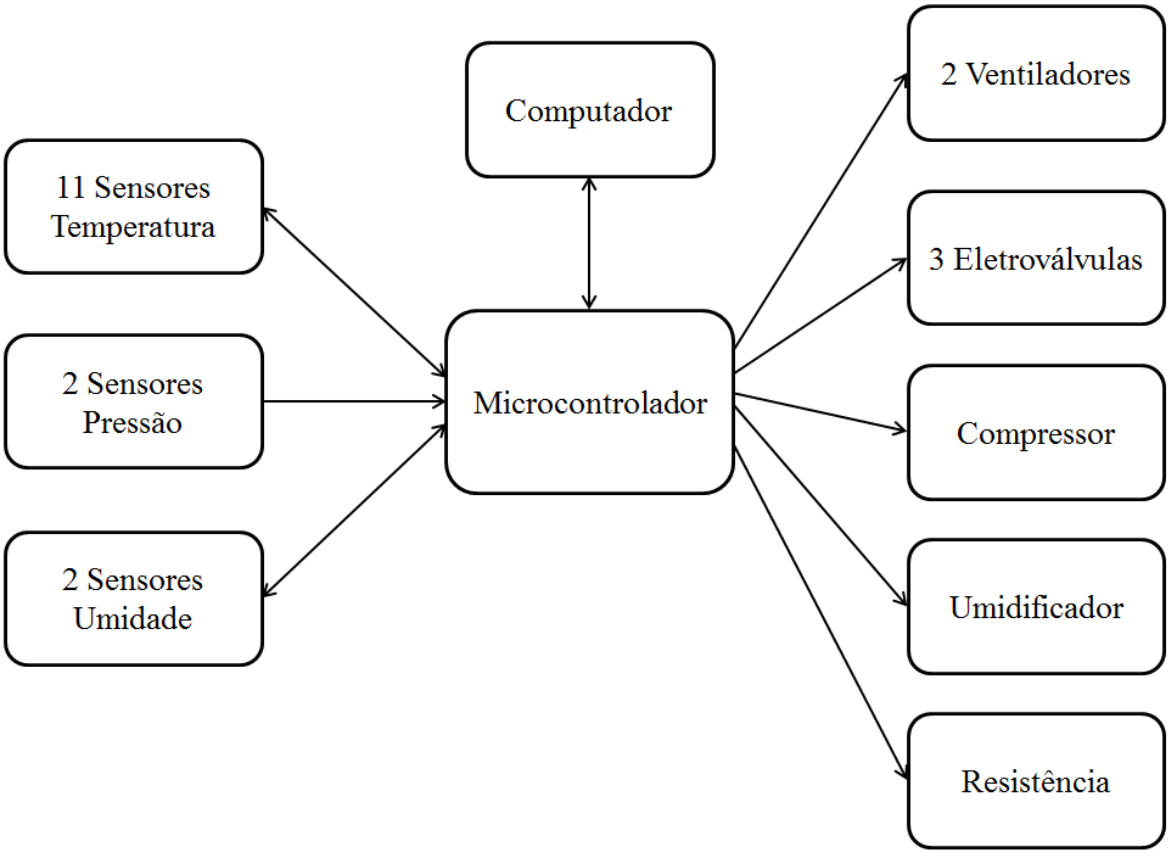
No capítulo 6 serão apresentadas as conclusões, assim como possíveis trabalhos futuros.

2. SISTEMA

2.1 Arquitetura Simplificada

O sistema será composto basicamente de um microcontrolador para controle, de sensores para realizar as medições, atuadores para acionamento da planta e de um computador para mostrar os dados e gerar os comandos de atuação. Na Figura 3 pode ser visto um diagrama de blocos simplificado dos componentes presentes no sistema.

Figura 3 - Diagrama de blocos do sistema.



Fonte: Próprio Autor

2.2 Arquitetura Detalhada

O microcontrolador usado será o STM32f103c8t6 com processador ARM® 32-bit Cortex® - M3, mostrado na Figura 4, responsável por ler os dados de sensores e atuar na planta a partir de comandos enviados pelo usuário. Dentre as alternativas comerciais este microcontrolador

se mostrou mais adequado ao projeto por ser o dispositivo com processador ARM com interface USB mais barato e fácil de encontrar no mercado brasileiro. Alternativas como o Tiva da Texas ou o Dischovery da própria STM, por exemplo, também atenderiam a demanda, porém são mais caros, maiores e com um excessivo número de periféricos, desnecessários ao projeto.

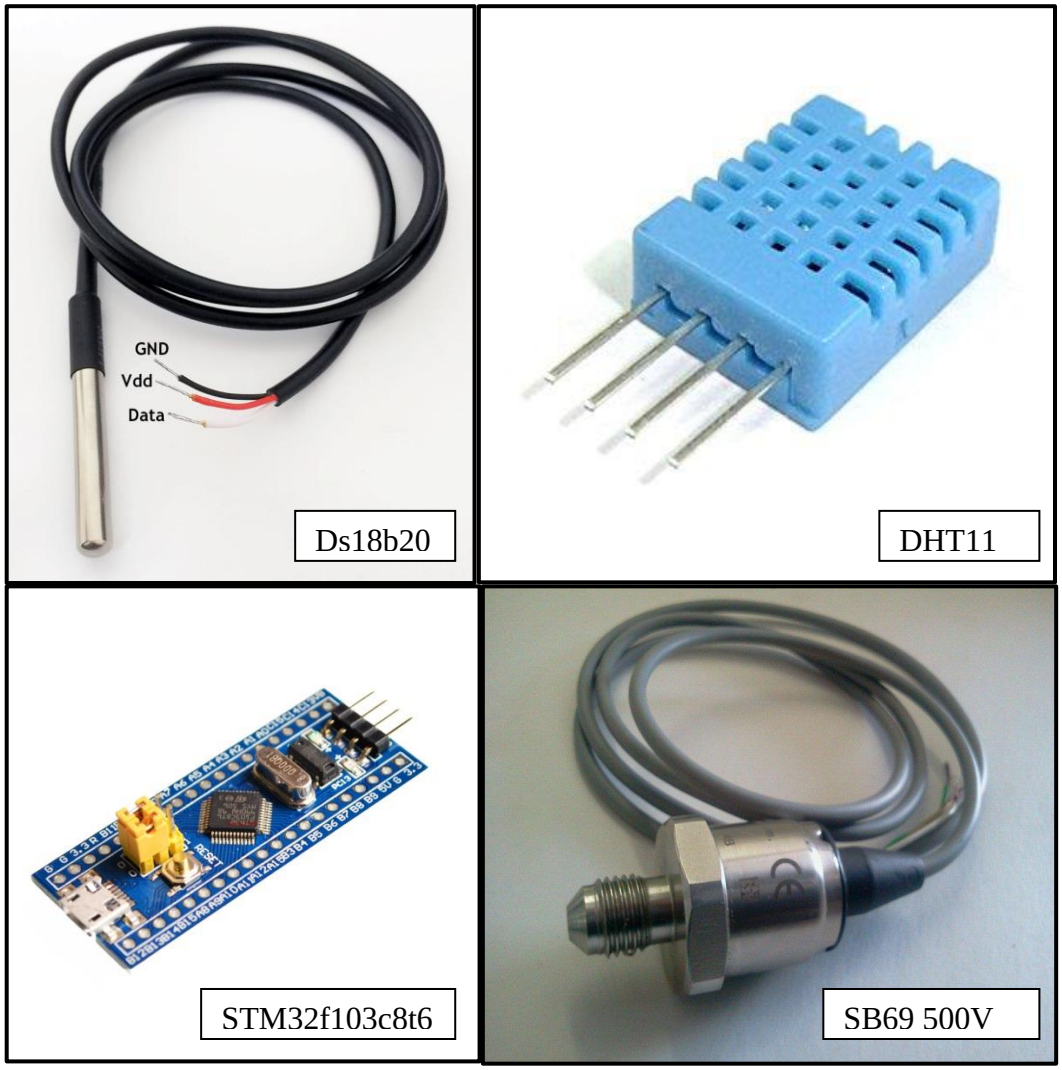
Para medir as temperaturas será utilizado o Ds18b20, um sensor digital que fornece informações a respeito da temperatura do dispositivo que podem variar de -10ºC a +85ºC, configurável para oferecer precisão de até 0,0625ºC.

Este sensor de temperatura foi escolhido para compor o projeto por, além de possuir as características necessárias, ser vendido já com um encapsulamento à prova d'água, como mostrado na Figura 4. Esta característica é crucial para o projeto visto que durante o processo de refrigeração ocorre congelamento parcial de algumas partes da bancada, além de acúmulos de água ao descongelar.

Outro motivo de sua escolha foi a capacidade de se colocarem vários sensores em um único barramento, o que possibilitaria a escalabilidade do sistema, conforme demandas futuras. Cada sensor possui um número de série exclusivo gravado em sua ROM e os dados são transmitidos a central microcontroladora a partir de um único fio utilizando o protocolo OneWire.

Para medir umidade será utilizado o DHT11, mostrado na Figura 4, um sensor também digital que se comunica por um protocolo próprio e fornece informações a respeito da umidade relativa do ar de 20% a 90%, com precisão de até 5%. Já a pressão será medida utilizando o Sb69 500V, um transdutor que gera sinais analógicos de tensão proporcionais a pressão aplicada, sendo capaz de trabalhar na faixa de 0 a 200 psi.

Figura 4- Sensores de temperatura (Ds18b20), umidade (DHT11) e pressão (SB69 500V) e microcontrolador (STM32f103c8t6) utilizados no projeto.



Fonte: Próprio Autor

A comunicação entre o microcontrolador e o computador será via USB, utilizando um protocolo criado pela própria Intechno Equipamentos Didáticos e que, por motivos da existência de um contrato de confidencialidade do autor deste projeto com a empresa, não será apresentado de maneira detalhada neste trabalho.

2.3 Arquitetura do *Software*

A escolha da estratégia a ser adotada na elaboração do *software* embarcado foi pautada na eficiência e confiabilidade na aquisição dos dados. Era necessário que a varredura dos

sensores fosse feita a uma taxa de até 300ms, visto que a interface computacional atualiza seus dados nessa faixa.

A resposta do microcontrolador aos comandos vindos da interface também deve ser realizadas de maneira confiável, visto que as estratégias de controle já estão implementadas e o mesmo só tem a função de responder corretamente aos comandos de acionamento.

Será utilizado o FreeRTOS na elaboração do *software*, dispondo de ferramentas que auxiliarão no controle dos tempos de leitura dos dados e atuação na planta. Este método foi escolhido por ser simples e confiável para o controle de tempo real, executando as funções de maneira determinística. Outro motivo de sua escolha foi de o mesmo também ser gratuito e sua licença permitir sua aplicação comercialmente.

O FreeRTOS funciona como uma camada de abstração entre o *hardware* e a Aplicação, tornando as aplicações em tempo real mais práticas, pois o usuário precisaria se preocupar somente com a funcionalidades do sistema por meio da criação de tarefas, deixando os detalhes do *hardware* para trás.

As tarefas podem assumir quatro estados:

- Pronta: a tarefa está pronta a entrar em execução, porém não está sendo executada;
- Em Execução: a tarefa está em execução;
- Bloqueada: a tarefa está esperando algum evento para continuar a sua execução;
- Suspensa: a tarefa foi suspensa pelo gerenciador de tarefas através da chamada de uma funcionalidade usada pala controlar as tarefas.

O FreeRTOS dispõe então de mecanismos para comunicação entre as tarefas, sendo eles filas de mensagens, semáforos e mutexes. As filas de mensagens funcionam como um canal de comunicação por onde as tarefas trocam mensagens, já os semáforos funcionam como habilitadores, permitindo somente a execução de tarefas que tiverem sinal verde para tal.

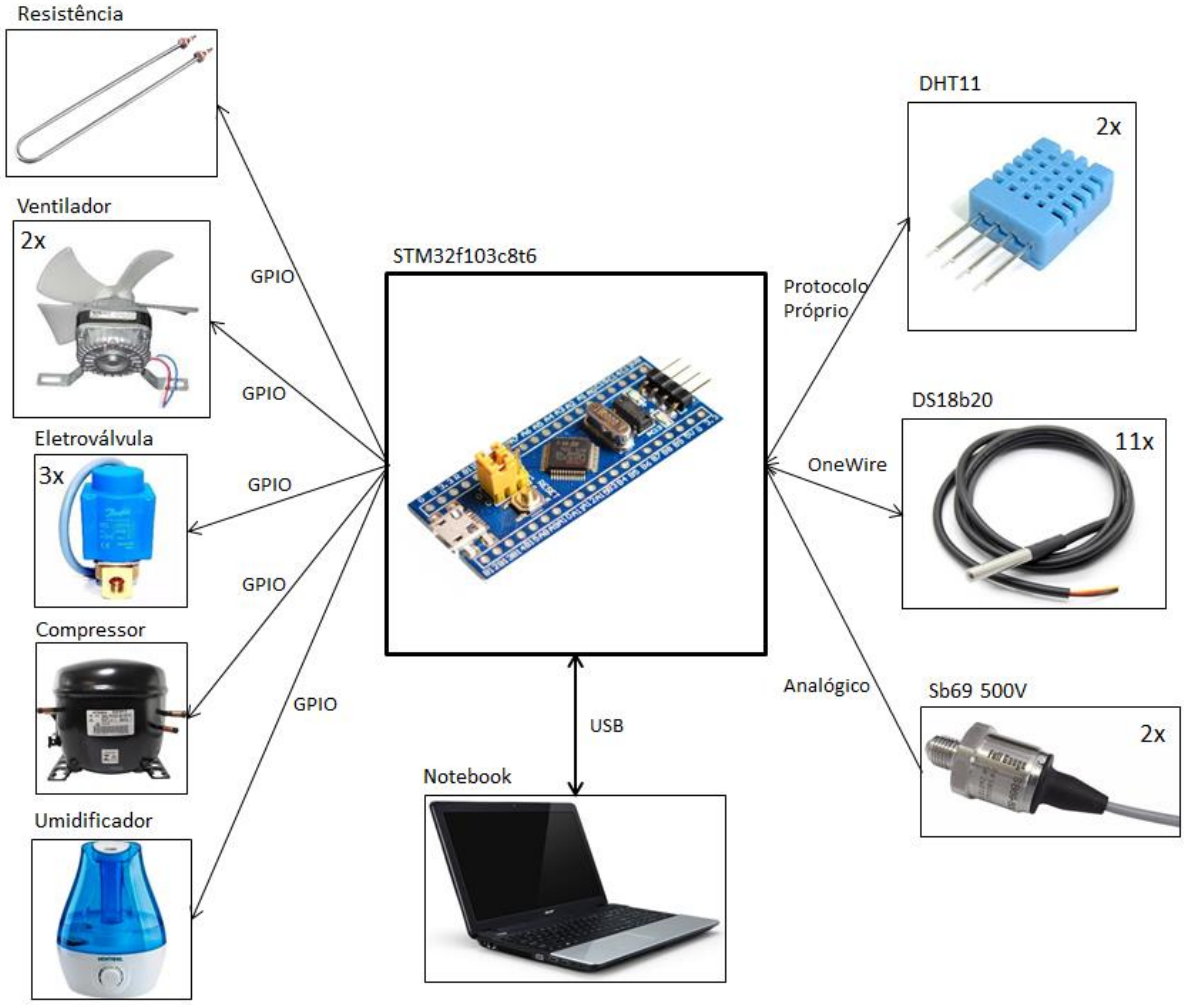
Os mutexes, por si só, funcionam de maneira semelhante aos semáforos, porém possuem um mecanismo de herança de prioridade, o qual impede que uma tarefa de menor prioridade bloqueie uma de maior, evitando assim bloqueio por inversão de prioridade.

2.4 Arquitetura do *Hardware*

Para cada periférico o STM32f103c8t6 utiliza uma forma diferente de interface. Para acionamento das cargas são utilizados os pinos GPIO configurados como saída. Já para os sensores de temperatura e umidade os pinos GPIO também são configurados como saída, porém com dreno aberto, visando executar protocolos de comunicação.

Nos sensores de pressão, a medição é feita utilizando um conversor A/D interno ao microcontrolador. Já a interface com o computador é feita utilizando comunicação USB, suportada pelo próprio microcontrolador, não sendo necessários conversores externos. Todos os periféricos assim como a comunicação utilizada estão mostrados na Figura 5.

Figura 5 - Esquemático dos periféricos utilizados.

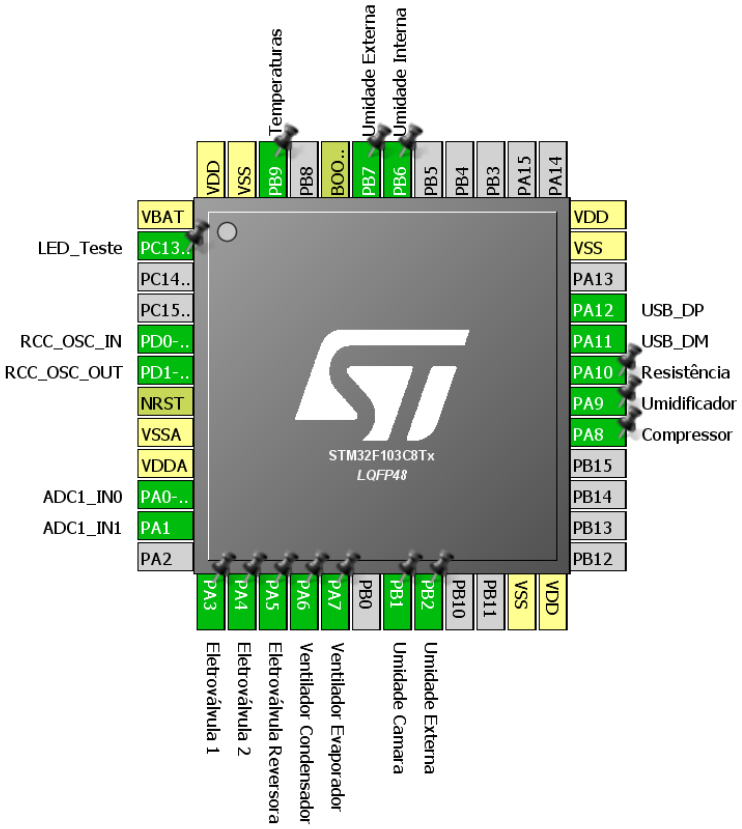


Fonte: Próprio Autor

3. IMPLEMENTAÇÃO DO *HARDWARE*

Primeiramente foram definidas as portas do microcontrolador que seriam usadas para leitura dos sensores e atuação nas cargas, como mostrado na Figura 6.

Figura 6 – Ligações no microcontrolador.



Fonte: Próprio Autor

Foi realizado então o projeto de alguns circuitos visando o acionamento das cargas pelo microcontrolador, visto que o mesmo é capaz de gerar na saída sinal lógico de 0 a 3,3V e suporta correntes de até 20mA. As cargas envolvidas no sistema estão descritas na Tabela 1.

Para realizar o acionamento dessas cargas foi utilizado o circuito retirado da folha de dados do MOC3023 e mostrado na Figura 7. Este circuito permite o desacoplamento do circuito CC com o circuito CA e possui capacidade de correntes de até 16A a uma tensão de 220V, suportando correntes de pico de até 140A, sendo suficientemente robusto para suportar as correntes de partida dos motores.

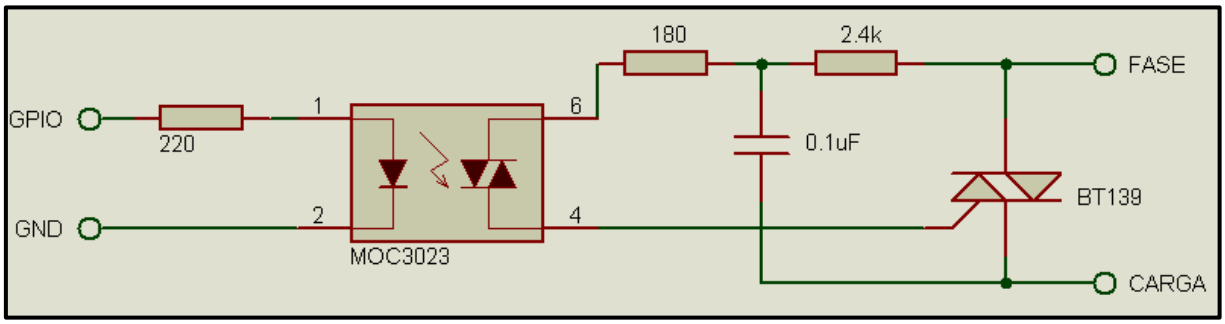
Tabela 1 – Características das cargas do sistema.

Componente	Tensão Nominal [V]	Corrente Nominal [A]
Eletroválvula	220	0,25
Compressor	220	3
Ventilador	220	1
Umidificador	220	1
Resistência	220	8

Fonte: Próprio Autor

Devido às características fortemente indutivas das cargas acionadas, é utilizado um circuito Snubber em paralelo com o Triac com o objetivo de protegê-lo contra picos de tensão, oriundos da variação instantânea da tensão na carga. O circuito Snubber também pode ser visto na Figura 7, sendo composto pelo capacitor de 0,1uF e o resistor de 2,4k em paralelo com o Triac.

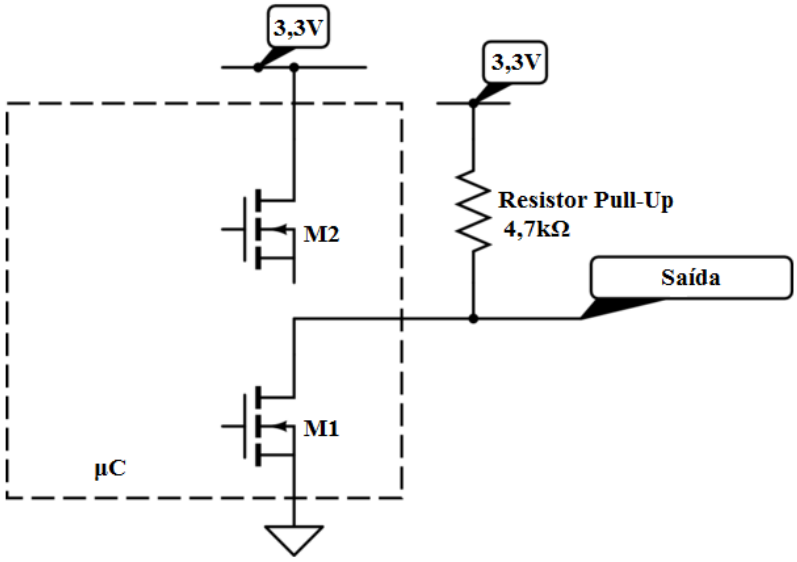
Figura 7 - Circuito de acionamento das cargas.



Fonte: Próprio Autor

Os sensores de temperatura e umidade necessitam de resistores de *pull-up* para o seu funcionamento. A porta do microcontrolador é configurada como saída com dreno aberto para realizar a comunicação com os sensores, utilizando o resistor de *pull-up* para obter o nível lógico alto, como mostrada na Figura 8.

Figura 8- Circuito de comunicação com os sensores de umidade e temperatura.



Fonte: Próprio Autor.

Para o sensor de pressão não foi utilizado nenhum circuito auxiliar, o tratamento do sinal foi realizado somente via *software*.

4. IMPLEMENTAÇÃO DO *SOFTWARE*

Primeiramente foi implementado um *software* auxiliar para realizar leituras individuais dos códigos ROM de 64 bits dos 11 sensores de temperatura utilizados no projeto, mostrados na Tabela 2. Esses dados foram então utilizados no *software* principal para identificação dos sensores no barramento de dados.

Tabela 2 – Códigos ROM dos 11 sensores de temperatura do sistema.

Sensor Ds18b20	**Código ROM**
Temperatura de entrada do condensador	0x28, 0xFF, 0xF0, 0xE1, 0x8C, 0x16, 0x03, 0x55
Temperatura de saída do condensador	0x28, 0xFF, 0x9F, 0xE0, 0x92, 0x16, 0x05, 0xC7
Temperatura de entrada do compressor	0x28, 0xFF, 0x7B, 0x59, 0x93, 0x16, 0x04, 0xF5
Temperatura de saída do compressor	0x28, 0xFF, 0xE7, 0xC5, 0x92, 0x16, 0x05, 0x65
Temperatura de entrada do evaporador	0x28, 0xFF, 0x3E, 0x26, 0xA0, 0x16, 0x04, 0xC6
Temperatura de saída do evaporador	0x28, 0xFF, 0x4F, 0xE2, 0x92, 0x16, 0x05, 0x9E
Temperatura do ar do condensador	0x28, 0xFF, 0x67, 0x13, 0x8B, 0x16, 0x03, 0xBF
Temperatura do compressor	0x28, 0xFF, 0x03, 0xE2, 0x93, 0x16, 0x05, 0xFD
Temperatura do ar do evaporador	0x28, 0xFF, 0x4E, 0x55, 0x93, 0x16, 0x04, 0xB1
Temperatura do interior da câmara	0x28, 0xFF, 0xC5, 0x72, 0x71, 0x16, 0x04, 0x74
Temperatura ambiente	0x28, 0xFF, 0xA2, 0x05, 0x83, 0x16, 0x05, 0x86

Fonte: Próprio Autor

A comunicação com os Ds18b20 foram feitas via protocolo OneWire, sendo necessário o conhecimento do protocolo na elaboração dos comandos de comunicação. Como a mesma é realizada de forma assíncrona, deve-se obedecer aos tempos descritos na documentação do protocolo. Informações a respeito do protocolo OneWire utilizado para o sensor pode ser visto no Anexo A.

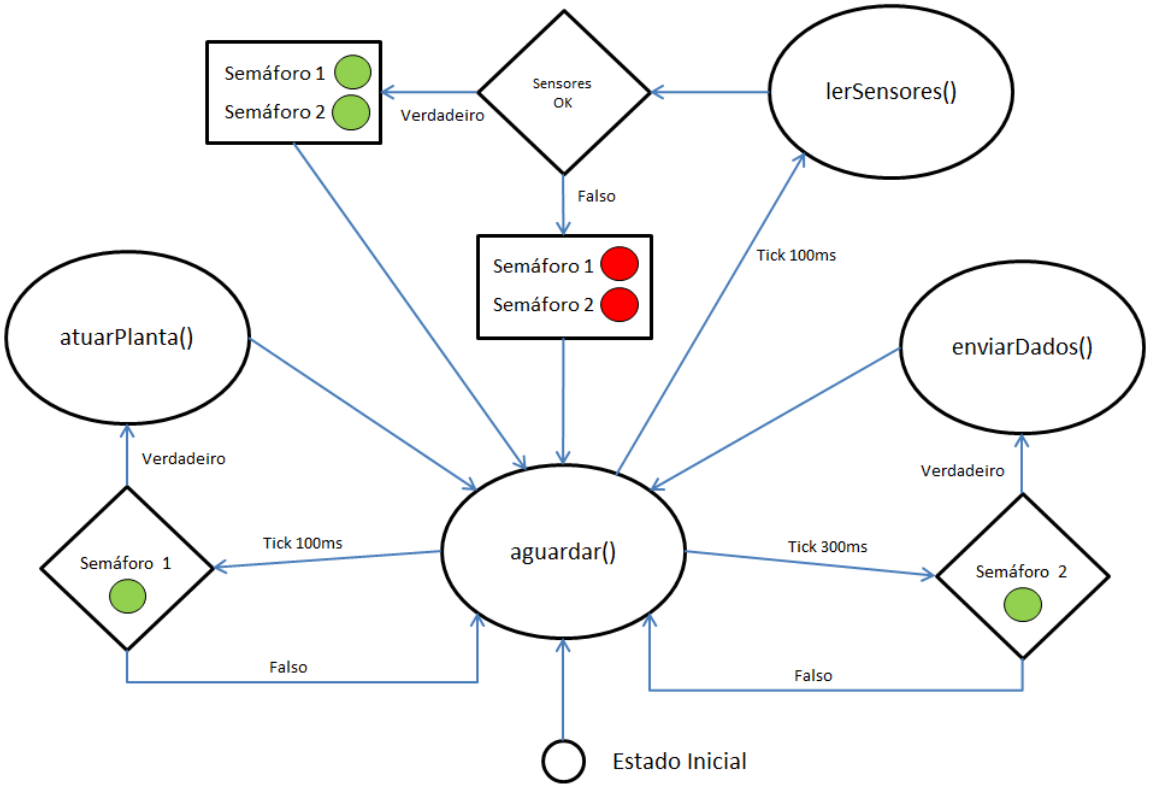
A comunicação com o DHT11 também é realizada de forma assíncrona via um protocolo proprietário, sendo necessário também um conhecimento aprofundado do mesmo para realizar a comunicação. Mais informações a respeito da comunicação com o DHT11 pode ser vistas no Anexo B.

Para medir pressão foram realizadas amostras analógicas do sinal e, via *software*, executada uma média móvel de três delas, funcionando como um filtro para o sinal.

Na elaboração do *software* foi utilizado o FreeRTOS e foram criadas três tarefas principais, sendo a de mais alta prioridade responsável pela leitura dos sensores (lerSensores()), a de prioridade mediana a responsável pela atuação na planta (atuarPlanta()) e a de baixa prioridade a de envio dos dados via USB (enviarDados()). A tarefa aguardar() é um estado de baixa energia, em que o microcontrolador aguarda o momento para que a próxima tarefa seja executada.

Como método de gestão das tarefas foram criados dois semáforos responsáveis pela habilitação das duas tarefas de menor prioridade, sendo o mesmo liberado pela tarefa de maior prioridade. Um diagrama simplificado das tarefas pode ser visto na Figura 9.

Figura 9- Diagrama simplificado das rotinas do sistema.



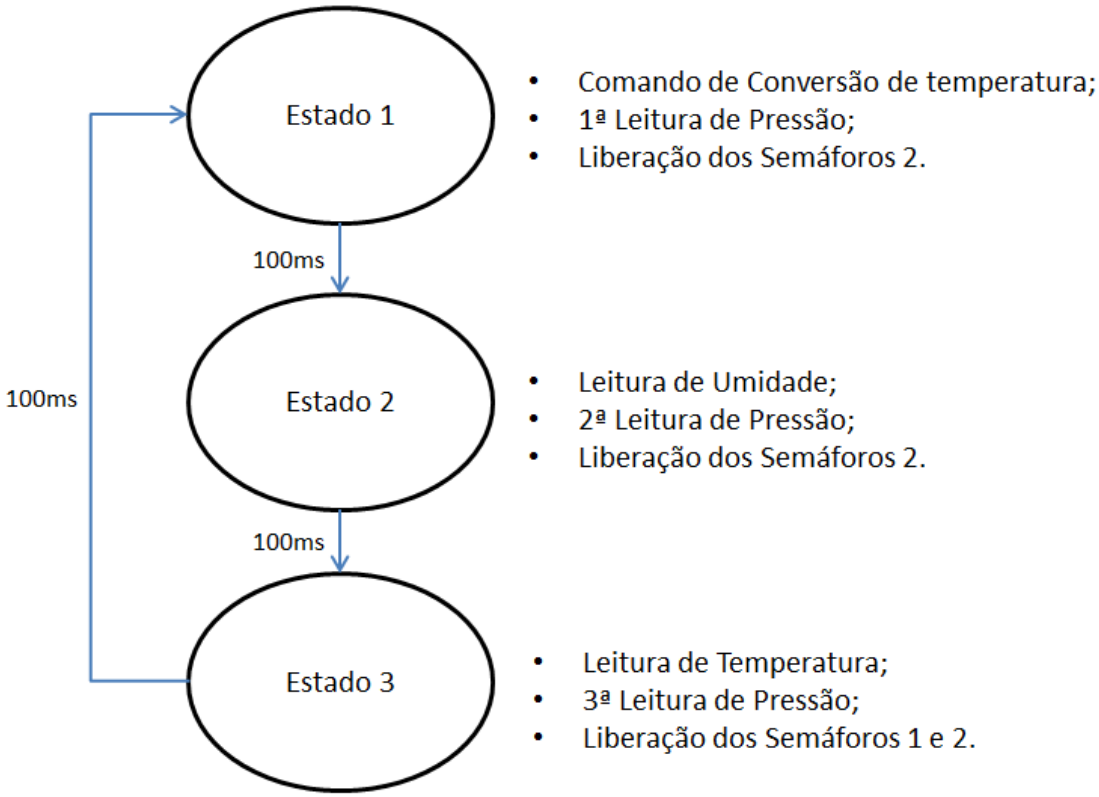
Fonte: Próprio Autor.

Interrupções geradas no meio do programa devido ao recebimento de pacotes via USB foram tratados de maneira rápida, colocando somente os dados em um buffer e os mesmos foram tratados dentro da rotina atuarPlanta().

Como o sensor Ds18b20 demora cerca de 100ms para realizar sua conversão de temperatura, a tarefa lerSensores() foi executada segundo uma máquina de estados, visando o melhor desempenho do sistema, sem perder sua característica de tempo real. A máquina de estados pode ser vista na Figura 10.

Com esse método, será possível garantir tempo real para o sistema, sendo os sensores atualizados a cada 300ms (após 3 estados de 100ms cada), os dados enviados também a cada 300ms e o microcontrolador atuando na planta a cada 100ms.

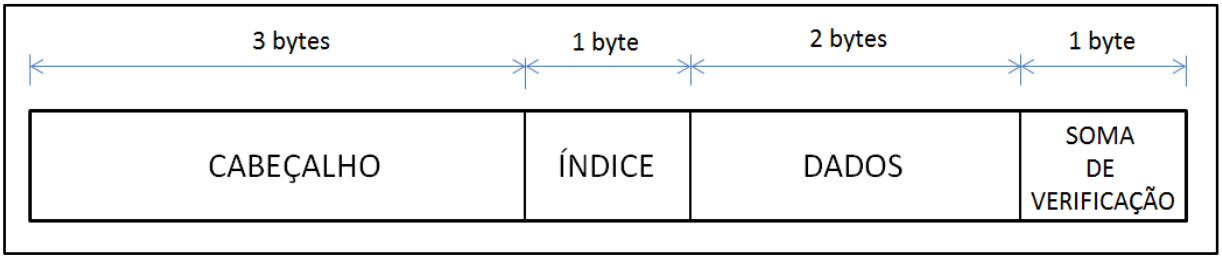
Figura 10- Estados da função lerSensores().



Fonte: Próprio Autor

A comunicação USB segue um protocolo desenvolvido pela empresa Intechno, realizado por meio do envio de pacotes com 7 *bytes* cada. Cada pacote possui um cabeçalho, um identificador para mostrar a qual variável o número se refere, o dado propriamente dito e um *byte* verificador. A estrutura do pacote pode ser vista na Figura 11.

Figura 11 - Estrutura do pacote de Dados



Fonte: Próprio Autor.

5. TESTES

Primeiramente foi realizado um teste de comunicação com o *software* via USB, enviando sinais aleatórios de temperatura, umidade e pressão. Este teste foi feito visando testar a comunicação com o *software* da empresa Intechno.

Foram realizados também testes individuais com os sensores de umidade, temperatura e pressão, visando obter o tempo de atualização dos mesmos. Os dados individuais do tempo aproximado de atualização e leitura para cada sensor individualmente estão mostrados na Tabela 3. Estes tempos são aproximados, pois, para medi-los, foi realizada uma análise dos dados vindas da USB, o que só era realizada a cada 10ms.

Tabela 3 – Tempos aproximados para atualização e leitura dos sensores.

Sensor	Tempo de Atualização e Leitura (ms)
Ds18b20	110
DHT11	3000
Sb69 500V	10

Fonte: Próprio Autor

Este teste mostrou que o sistema obtém um dado atualizado de umidade a cada 3000ms, isso se dá devido ao tempo de amostragem do sensor que, como mostrado na própria folha de dados, é superior a 2 segundos.

Os tempos de leitura dos sensores de temperatura, apesar de demorar cerca de 110ms para um único sensor, não são críticos. Essa não criticidade se dá, pois, o maior tempo gasto é referente à conversão da temperatura, que é 93,75ms segundo a folha de dados do sensor, o que é feita de maneira simultânea aos 11 Ds18b20 do barramento.

O sensor de pressão possui uma atualização mais rápida, visto que é um sinal analógico. O tempo necessário a atualização é composto principalmente pelo tempo de conversão A/D do STM32f103c8t6 e do filtro realizado no próprio *software*.

Foram utilizados sensores externos com o objetivo de aferir os valores de temperatura, umidade e pressão medidos. Para a aferição da pressão foi utilizados um manômetro presente

na própria bancada de refrigeração. Já a temperatura e a umidade foram aferidas utilizando um sensor infravermelho e um termo higrômetro digitais respectivamente. A comparação dos valores de um teste realizados com os sensores externos e os do projeto em questão estão mostrados na Tabela 4.

Tabela 4 – Comparação entre os sensores utilizados no projeto e sensores comerciais.

Sensor	Valor	Sensor	Valor	Erro
Ds18b20	25.6	Sensor Infravermelho Digital	25.3	1,18%
DHT11	69%	Termo higrômetro Digital	75%	8%
Sb69 500V	152psi	Manômetro Analógico	155psi	1,93%

Fonte: Próprio Autor

O terceiro teste realizado foi o de acionamento das cargas, mostrando a eficácia do circuito projetado no acionamento das mesmas, sem apresentar nenhum tipo de superaquecimento ou falha.

Para o teste final foi realizado o controle de temperatura no interior da câmara, como se trata de um controle ON/OFF, quando foi atingida a temperatura mínima imposta pelo usuário, o compressor desligou, mantendo somente ligada a eletroválvula do ciclo escolhido e o ventilador do evaporador, como esperado. Quando a temperatura atingiu novamente o valor estipulado para religamento, o sistema voltou a funcionar normalmente. Os valores mostrados na interface gráfica podem ser vistos na Figura 12.

Figura 12 – Dados dos sensores mostrados na interface gráfica.

Fonte: Próprio Autor

6. CONCLUSÕES E TRABALHOS FUTUROS

Os testes mostram que o sistema funciona corretamente, lendo precisamente os dados dos sensores e atuando na planta como esperado. A comunicação com o *software* também foi validada, visto que as informações dos sensores na tela do computador ocorrem no tempo previsto.

Este trabalho foi de grande valia no aprimoramento de conceitos de protocolos de comunicação, microcontroladores com processadores ARM, sistemas de tempo real, circuitos de acionamento de cargas e refrigeração.

Para trabalhos futuros podem ser adicionados novos sensores de temperatura ao barramento, o que é totalmente suportado pelo sistema. O mesmo também possui portas digitais e analógicas sobrando, que podem ser utilizadas para outros sensores, além dos previamente descritos. Outras cargas também podem ser adicionadas a planta, como uma resistência simulando cargas térmicas no interior da câmara, por exemplo.

O projeto de uma PCI também deverá ser realizado caso se queira embarcar o projeto aqui descrito em uma Bancada de Refrigeração comercializável.

REFERÊNCIAS

SADASIVAN, S. **An Introduction to the ARM Cortex-M3 Processor** 2006. White Papper to ARM.

[FOLHA DE DADOS DHT11, 2017] Aosong(Guangzhou) Electronics Co.,Ltd, **Temperature and humidity module DHT11 Product Manual**, Folha de Dados do DHT11. Disponível em: https://akizukidenshi.com/download/ds/aosong/DHT11.pdf. Acesso em 10 de março de 2017

[FOLHA DE DADOS DS18B20, 2017] Dallas Semiconductor, **DS18B20 Programmable Resolution 1-Wire® Digital Thermometer**, Folha de Dados do Ds18b20. Disponível em: https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf em 10/07/2017. Acesso em 10 de março de 2017

STMicroelectronics, **STM32F101xx, STM32F102xx, STM32F103xx, STM32F105xx and STM32F107xx advanced ARM®-based 32-bit MCUs**, Folha de Dados do STM32f103c8t6 [Revisão 26/11/2015]. Disponível em: http://www.st.com/content/ccc/resource/technical/document/reference_manual/59/b9/ba/7f/11/af/43/d5/CD00171190.pdf/files/CD00171190.pdf/jcr:content/translations/en.CD00171190.pdf. Acesso em: 21 de abril de 2017

STMicroelectronics, **How to get the best ADC accuracy in STM32 microcontrollers**, Folha de Dados [Revisão 15/02/2017]. Disponível em: http://www.st.com/content/ccc/resource/technical/document/application_note/group0/3f/4c/a4/82/bd/63/4e/92/CD00211314/files/CD00211314.pdf/jcr:content/translations/en.CD00211314.pdf. Acesso em: 22 de abril de 2017

[INTECHNO, 2017] Intechno Equipamentos Didáticos, **MANUAL DO PRODUTO - PLANTA DE REFRIGERAÇÃO**, Manual de utilização [Revisão 07/07/2017].

[TERMODINÂMICA APLICADA, 2017] **Ciclo de Refrigeração**. Disponível em : http://www.ebah.com.br/content/ABAAABQTYAJ/sistemas-refrigera-uo-por-compress-uo-vapor. Acesso em: 22 de junho de 2017.

The FreeRTOS™ Reference Manual. Disponível em: http://www.freertos.org/Documentation/FreeRTOS_Reference_Manual_V9.0.0.pdf. Acesso em: 11 de junho de 2017.

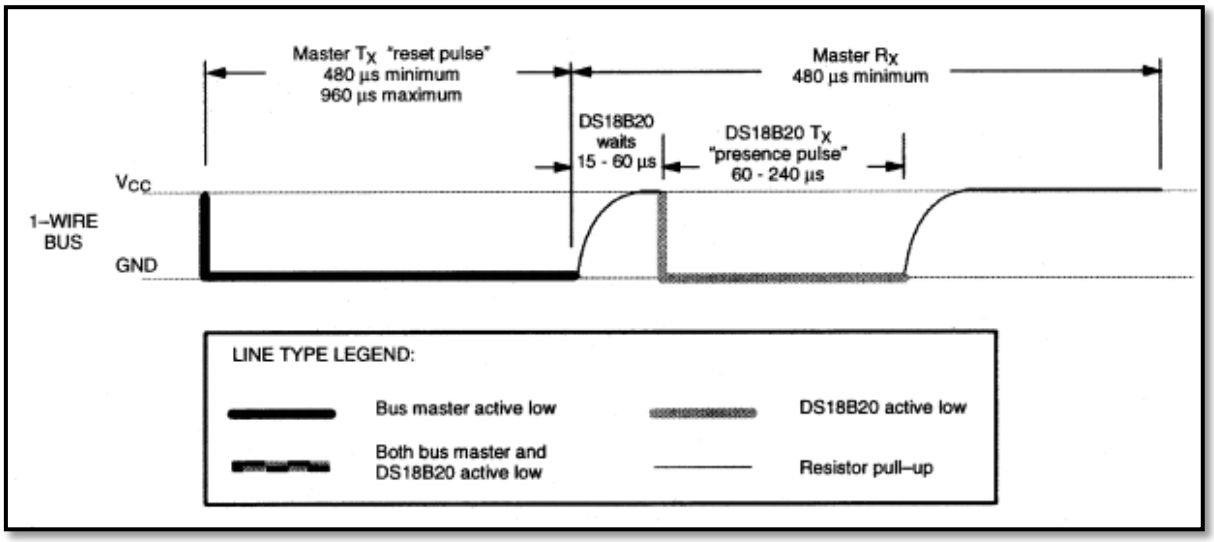
Mastering the FreeRTOS™ Real Time Kernel. Disponível em: http://www.freertos.org/Documentation/161204_Mastering_the_FreeRTOS_Real_Time_Kernel-A_Hands-On_Tutorial_Guide.pdf. Acesso em 22 de junho de 2017.

ANEXO A

Sensor Ds18b20 Comunicação OneWire

Toda transação no barramento começa com uma sequência de inicialização como mostrada na Figura 13. A mesma consiste de um pulso de *reset* transmitido pelo microcontrolador seguido de um pulso de presença enviado pelo sensor, indicando que o mesmo encontra-se preparado para operar. Os tempos dos pulsos para inicialização podem ser vistos na Figura 13.

Sequencia de inicialização do Ds18b20.

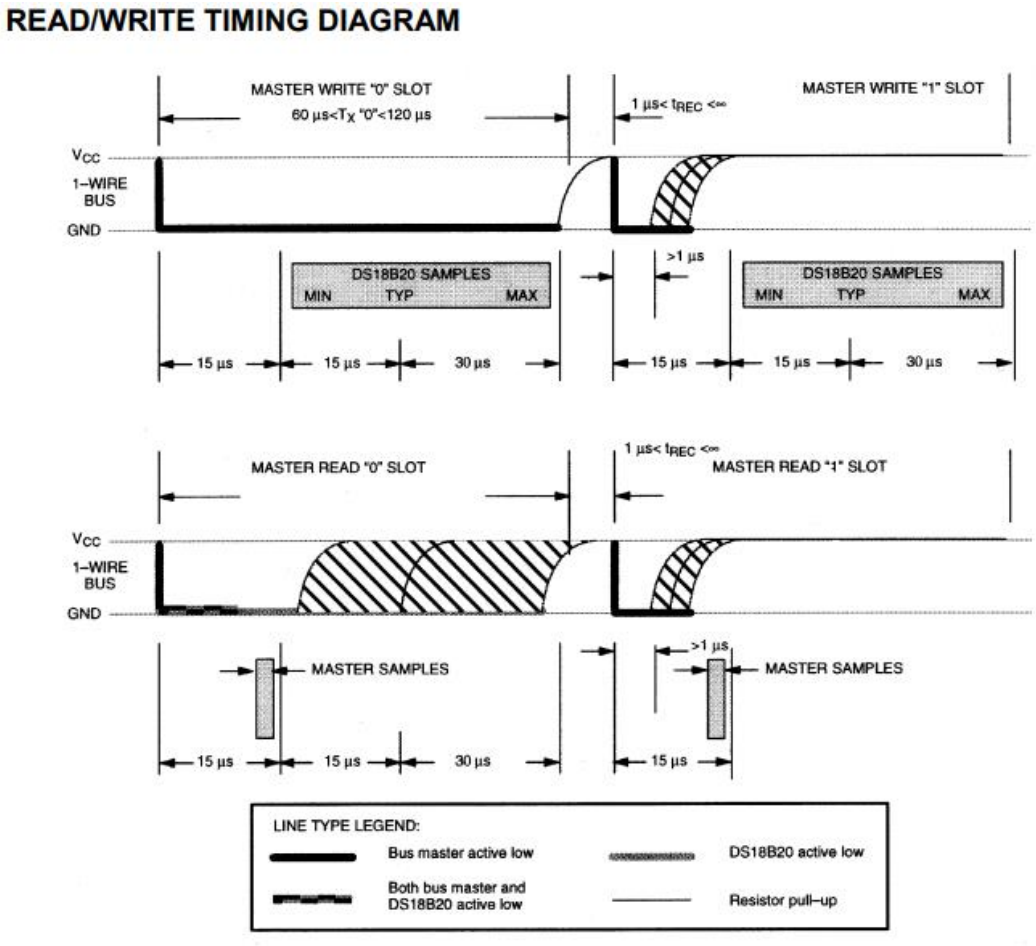


Fonte: FOLHA DE DADOS DS18B20, 2017

Para realizar a escrita para o sensor é necessário primeiramente gerar um pulso de *reset* com o microcontrolador seguido de um nível lógico alto ou baixo, dependendo do bit a ser enviado. Já para leitura ocorre também um pulso de *reset*, seguido da amostragem cerca de 15µs após o pulso. A temporização da leitura e da escrita está mostrada na Figura 14.

Existem dois tipos de comando a serem enviados ao sensor, os Comandos da ROM e os Comandos de Memória. Ambos são comandos de um *byte*, cada um referente a uma operação a ser executada no Ds18b20.

Diagrama de leitura e escrita de dados com o Ds18b20.



Fonte: FOLHA DE DADOS DS18B20, 2017

Comandos da ROM

Os comandos ROM possuem 8 bits cada e só podem ser dados após a inicialização. Os cinco comados ROM são:

Read ROM [33h]: este comando permite que o microcontrolador realize a leitura dos 64 bits da ROM e somente pode ser usado quando só tem um sensor Ds18b20 no barramento. Estes bits são referentes à família, o número de série e o CRC do sensor;

Match ROM [55h]: este comando, seguido do código da ROM de 64 bits, permite que o microcontrolador enderece um Ds18b20 específico no barramento;

Skip ROM [CCh]: este comando é utilizado para economizar tempo, permitindo que o microcontrolador envie funções de memória sem que tenha enviado o código ROM de 64 bits específico do sensor. Como não se envia o código ROM de 64 bits não se pode utilizar este comando com mais de um sensor no barramento, senão haverá colisão de dados;

Search ROM [F0h]: este comando é utilizado para que o microcontrolador realize o mapeamento de todos os Ds18b20 do barramento, identificando os códigos ROM de 64 bits dos sensores. A identificação do sensor pelo microcontrolador leva cerca de 13,16ms para acontecer, sendo o mesmo capaz de identificar 75 sensores no barramento por segundo;

Alarm Search [ECh]: este comando é utilizado para verificar em quais sensores ocorreram temperaturas maiores que TH (limite superior de temperatura setado para alarme) ou menores que LT (limite inferior de temperatura setado para alarme). Só irão responder a este comando sensores que tiveram o bit de alarme setado.

Comandos de memória

Os comandos de memória, assim como os comandos da ROM, também apresentam 8 bits cada e só podem ser utilizados após executado algum comando de ROM. Os 6 comandos de memória são:

Write Scratchpad [4Eh]: este comando escreve 3 *bytes* na *scratchpad memory* do Ds18b20 referentes a TH, TL e configuração;

Read Scratchpad [BEh]: este comando realiza a leitura dos 8 *bytes* da *scratchpad memory*, iniciando do *byte* 0 até 8, podendo a leitura ser interrompida a qualquer momento pelo mestre a partir de um pulso de *reset*;

Copy Scratchpad [48h]: este comando realiza a copia dos valores de TH, TL e configuração da *scratchpad memory* para a EEPROM;

Convert T [44h]: este comando indica para o Ds18b20 que realize a conversão de temperatura;

Recall E2 [B8h]: este comando copia dos dados de TH, TL e configuração da EEPROM para a *scratchpad memory*. Este processo de copia também é executado automaticamente toda vez que o sensor é reiniciado;

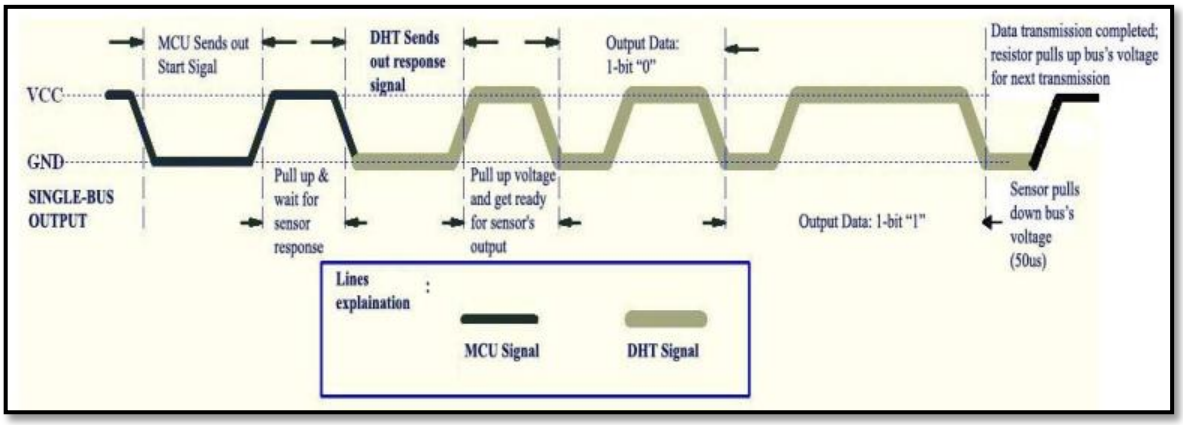
Read Power Supply [B4h]: este comando permite que o microcontrolador saiba qual modo de alimentação está sendo utilizado pelo sensor, sendo 0 o modo parasita e 1 o modo de alimentação externa.

ANEXO B

Sensor DHT11 Comunicação

A troca de dados entre o sensor e o microcontrolador é realizada a partir de um único fio, como visto na Figura 15 e se inicia com o microcontrolador enviando um sinal de nível lógico baixo superior a 18ms. Após enviar o sinal de inicio, o microcontrolador aguarda a resposta do sensor, que sairá do estado de espera e enviará também um sinal de nível lógico baixo, dando inicio assim ao envio dos dados de umidade, temperatura e do bit de paridade. Com a transmissão completa, o DHT11 envia um sinal de nível lógico baixo por um tempo superior a 50µs, dando fim a transmissão e se colocando novamente em um estado de espera.

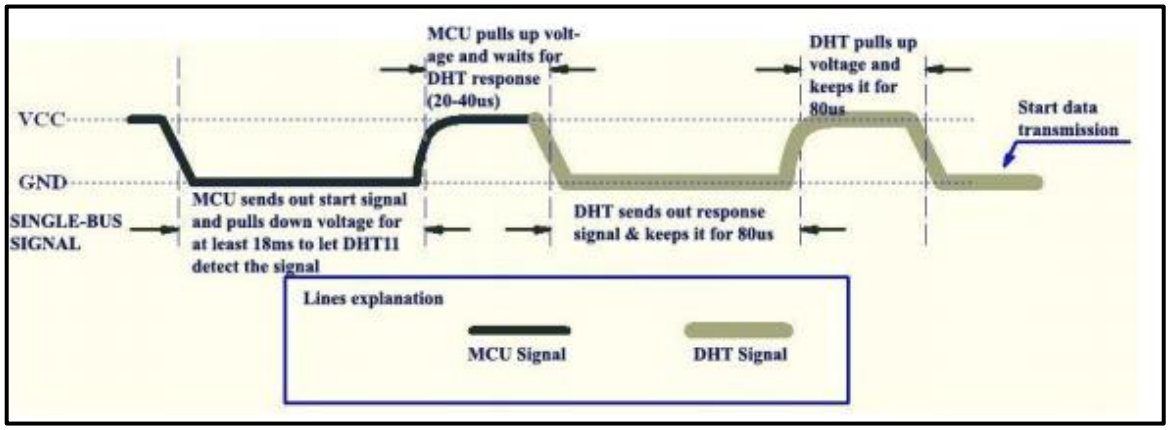
Comunicação de dados do microcontrolador com o DHT11.



Fonte: FOLHA DE DADOS DHT11, 2017

O inicio da transmissão está mostrado na Figura 16, começando com o microcontrolador enviando um sinal de nível lógico baixo por 18ms e após colocando-se em estado de espera, aguardando um sinal de nível lógico baixo do sensor de 20 a 40µs, indicando que o mesmo está preparado e saiu do modo de baixa energia. Antes de iniciar a transmissão o sensor ainda fica em nível lógico baixo por 80µs e depois espera mais 80µs para começar a transmissão dos 8 *bytes*.

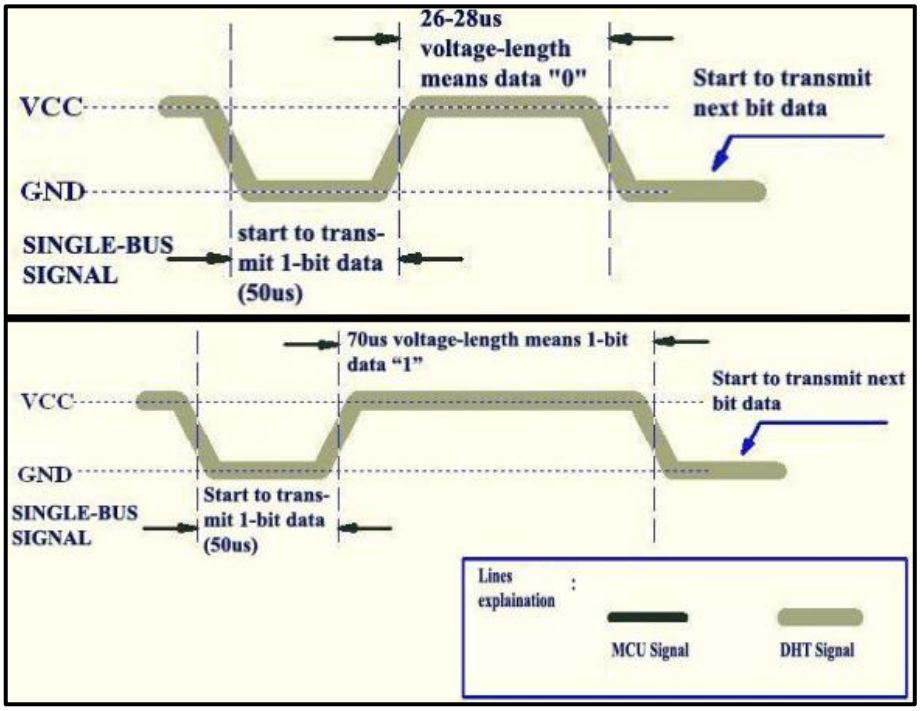
Início da comunicação do microcontrolador com o DHT11.



Fonte: FOLHA DE DADOS DHT11, 2017

Nos dados, todo bit inicia-se com um sinal de nível lógico baixo de 50µs, sendo o comprimento do pulso seguinte em nível lógico alto o determinante para o bit ser ‘0’ ou ‘1’, como mostrado na Figura 17.

Representação dos bits ‘0’ e ‘1’ pelo DHT11.



Fonte: FOLHA DE DADOS DHT11, 2017