

UNIVERSIDADE FEDERAL DE SANTA MARIA

CENTRO DE TECNOLOGIA

ENGENHARIA DE COMPUTAÇÃO

SENSORIAMENTO DE TEMPERATURA E

CONTROLE DE REFRIGERAÇÃO

USANDO FreeRTOS NO ARDUINO

RELATÓRIO DA DISCIPLINA DE

PROJETO DE SISTEMAS EMBARCADOS

Prof. Carlos Henrique Barriquello

Franciúne Barbosa da Silva de Almeida

Victor Eugenio Mainardi Fritz

Santa Maria, 26 de Agosto

2021

RESUMO

O presente documento tem como objetivo relatar o processo de desenvolvimento do projeto final da disciplina, que consiste na simulação de um sensor de temperatura baseado em um sistema operacional em tempo real. Para este, foi utilizado o software de simulações elétricas e eletrônicas *Proteus*, integrando o uso da simulação de uma placa *Arduino UNO* e dos demais componentes do circuito. Além da própria IDE Arduino para a implementação da mesma.

ÍNDICE

1 INTRODUÇÃO	3
1.1 FUNDAMENTAÇÃO TEÓRICA	3
2 DESENVOLVIMENTO	3
2.1 MATERIAIS UTILIZADOS	4
2.2 HARDWARE	4
2.3 SOFTWARE	6
3 IMPLEMENTAÇÃO EMBARCADA	7
4 MODELAGEM DO SISTEMA EMBARCADO	8
REFERÊNCIAS BIBLIOGRÁFICAS.....	10

1 INTRODUÇÃO

Através de três tarefas executadas independentemente, temos o controle de um ar condicionado através do sensoramento de temperatura, cuja leitura é realizada através de um sensor que, posteriormente, envia os dados para serem calculados com o objetivo de realizar ações nos atuadores do sistema.

1.1 FUNDAMENTAÇÃO TEÓRICA

O Arduino Uno é uma placa de Arduino que tem como microcontrolador principal o ATmega328P da fabricante Atmel. Ele possui 14 pinos digitais que podem ser utilizados como entrada e/ou saída. Dentre eles, 6 pinos podem ser utilizados como saída PWM, que é um tipo de sinal elétrico para controle de motor por largura de pulso, e 6 pinos de entrada para sinais analógicos. Para o clock do microcontrolador, é utilizado um cristal oscilante de 16Mhz. Ele possui uma conexão USB e um conector para ligação da fonte de energia, um conector para programação e um botão de reset para inicialização da placa.

A biblioteca de código livre FreeRTOS oferece um sistema operacional em tempo real para sistemas embarcados. Por ser de código aberto (“open-source”), isso permitiu o surgimento de várias versões do FreeRTOS que suportam vários dispositivos. Atualmente, existem mais de 35 versões de sistemas operacionais da série de processadores, incluindo Atmel AVR, cujo microcontrolador ATmega328 é o principal componente do Arduino ONU.

O sensor LM35 é um sensor de precisão que apresenta uma saída de tensão linear em relação à temperatura, assim que for alimentado. Seu terminal de saída emite um sinal de 10mV por grau Celsius. Ele se sobressai em relação a outros sensores quando consideramos esta medida de temperatura, já que a maioria dos dispositivos de sensoramento trabalham com a escala Kelvin, assim, o LM35 tem uma saída mais precisa, visto que nenhuma variável é subtraída, além de ter um custo reduzido para o sistema.

2 DESENVOLVIMENTO

2.1 MATERIAIS UTILIZADOS

Para um sistema ser embarcado, no entendimento básico e objetivo dos conhecimentos adquiridos ao longo da disciplina, ele é definido como a integração de um software que é modelado para o funcionamento de um hardware, que tem como objetivo fazer essa leitura e executar determinadas ações, como por exemplo sensores e atuadores para um problema específico.

O uso de sistemas embarcados cresce diariamente devido a necessidade de sistema auxiliar para um determinado problema, que, humanamente, é inviável e ineficiente realizar. Para essa integração, é necessário a modelagem da lógica de um programa computacional “software” que seja embutido “embarcado” em um dispositivo físico “hardware”. Conclui-se, que para um sistema ser embarcado, precisa de uma forte relação e integração entre hardware e software.

- Placa de desenvolvimento Arduino UNO;
- Display LCD 16x2 para monitoramento do estado do sistema;
- LED's verde e vermelho para sinais de alerta do sistema;
- Sensor de temperatura LM35 para leitura de sinal analógico;
- Simulador Tinkercad para simulação do problema;
- Simulador Proteus para simulação do sistema embarcado completo.

2.1 HARDWARE

Para complementar o desenvolvimento do software e realizar a adaptação do projeto para um projeto de sistema embarcado, é necessário e basicamente obrigatório termos um sistema de hardware para comportar e servir de alicerce para este desenvolvido, bem como funções específicas de um sistema de tempo real.

Foi utilizado então, a placa Arduino UNO, que faz leitura do input analógico de um sensor de temperatura LM35, e também os atuadores do sistema, LED's que informam a situação/estado desse sistema e também um motor que simula um sistema de refrigeração.

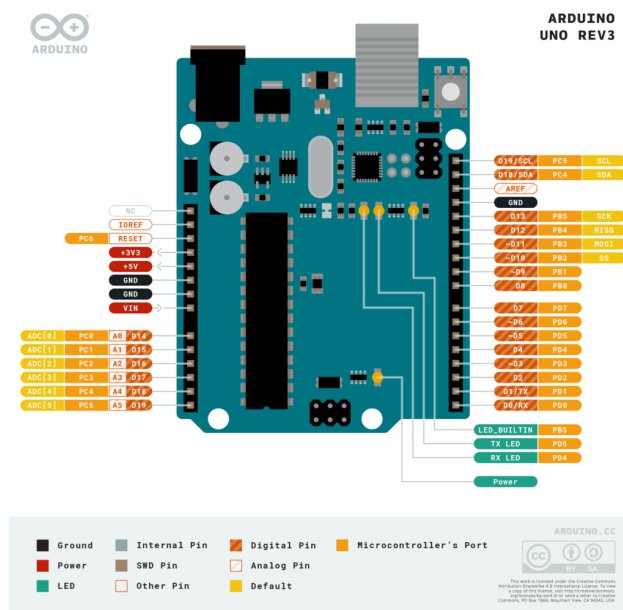


Figura 2.1.1 - Diagrama Arduino Mega 2560

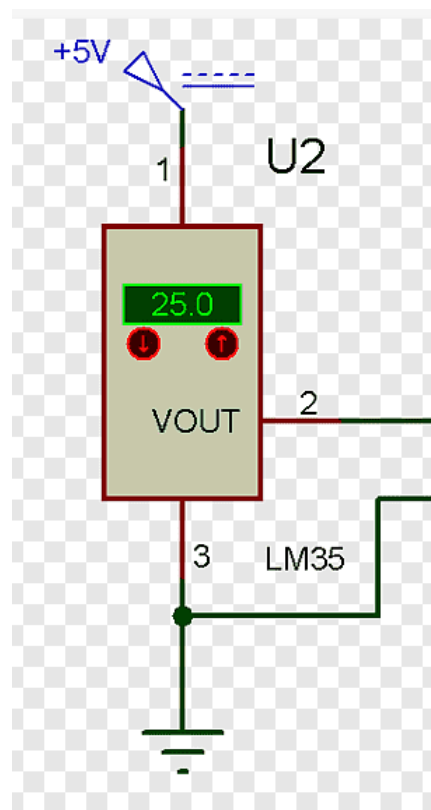


Figura 2.1.2 - Diagrama Sensor LM35 no Proteus

2.2 SOFTWARE

Para a modelagem do software, foi utilizado o ambiente de desenvolvimento Arduino IDE, foi também necessário fazer o uso de uma biblioteca específica de funções, a *Arduino_FreeRTOS*, biblioteca responsável por fazer a manipulação de funções e conceitos de um Sistema Operacional de Tempo Real (*RTOS, do inglês*). Esse RTOS mostra a necessidade, no momento em que um sistema opere todo o tempo “full-time”, para que ele não obtenha imprecisão e falha de comunicação na leitura desses dados, trazendo resultados precisos e leais.

O desenvolvimento das funções RTOS no software foi modelado através do ambiente de desenvolvimento Arduino IDE, que após a importação dessas bibliotecas foi possível escrever funções que usassem o conceito de Semáforos, Mutex, Ticks, entre outras.

3 IMPLEMENTAÇÃO EMBARCADA

A alternativa para que a adaptação do projeto de hardware suportasse o desenvolvimento do software, foi o simulador elétrico e eletrônico *Proteus* (figura 3.1), que apresentou diversas falhas, erros, durante a implementação. Devido a ele ser software privado, a versão gratuita também possui limitações, e as alternativas de obtenção de métodos gratuitos para o uso dele são totalmente desaconselhadas.

Analisando o ambiente de desenvolvimento do projeto, foi identificado que a plataforma de simulação de hardware *Tinkercad* (figura 3.2) não possuía suporte para bibliotecas de RTOS, pelo fato que ela é uma plataforma gratuita e limitada.

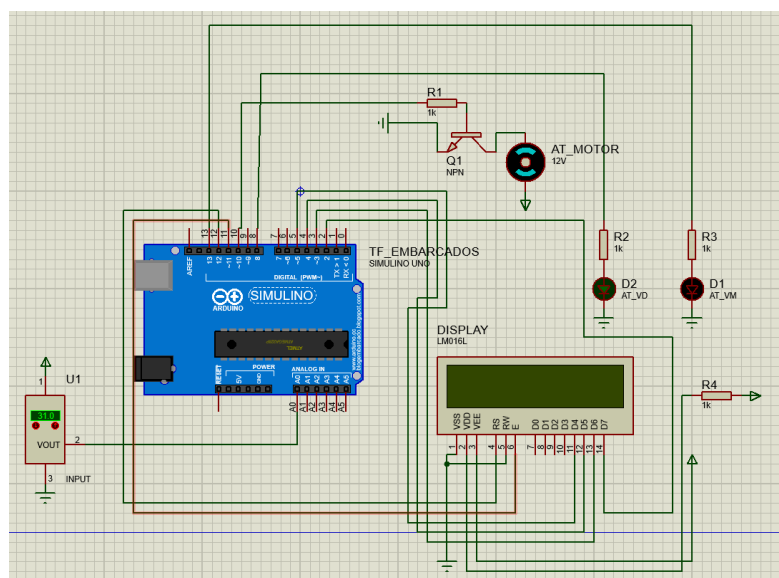


Figura 3.1 - Simulação eletrônica no Proteus

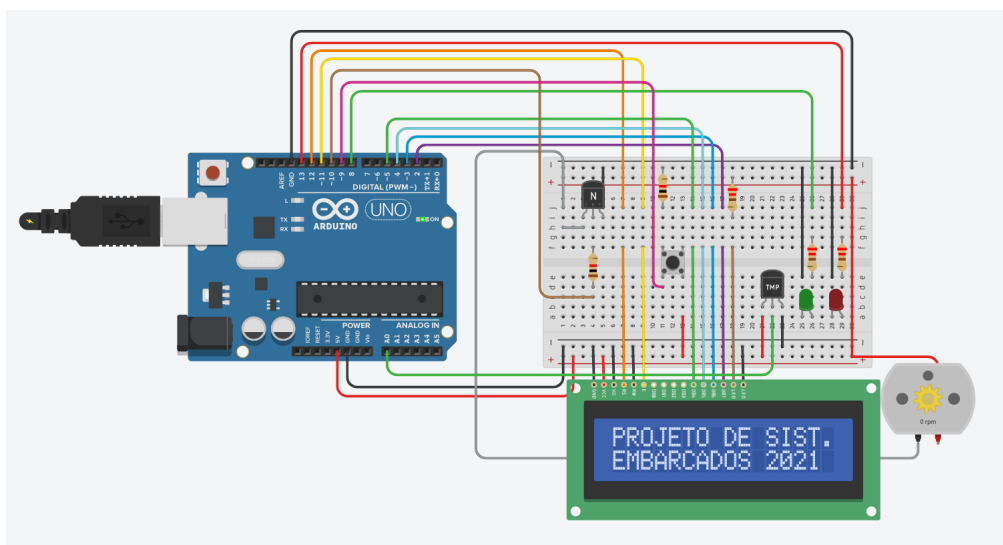


Figura 3.2 - Simulação eletrônica no Tinkercad

4 MODELAGEM DO SISTEMA EMBARCADO

O sistema faz a leitura dos dados analógicos através do sensor de temperatura LM35, posteriormente, codifica essas informações convertendo-as em níveis de tensão e graus celcius. Após isso, compara se as temperaturas atingiram determinado valor, para acionar os atuadores do sistema que fará o controle da refrigeração e sinais de alerta.



Figura 4.1 - Representação do sistema em alto nível

Os sinais de alerta e atuadores, são acionados somente nas seguintes situações:

Se a temperatura média for maior que 32°C, aciona refrigeração;

Se a temperatura média for menor que 20°C, desligue a refrigeração;

Se a temperatura média for maior que 28°C, aciona o sinal de alerta vermelho, indicando possível acionamento da refrigeração;

Se a temperatura média for menor que 24°C, aciona o sinal de alerta verde, indicando estabilidade na temperatura e possível desligamento da refrigeração.

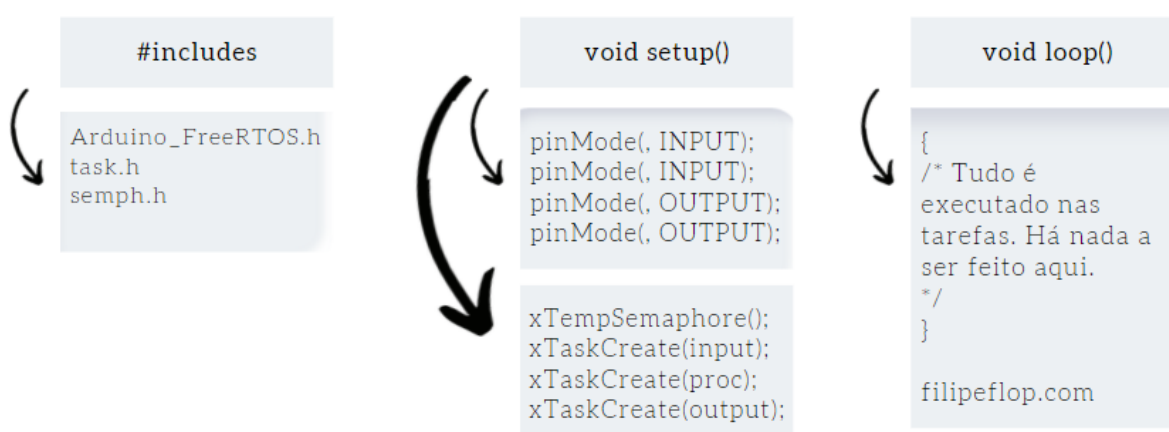


Figura 4.2 - Estrutura das funções Setup e Loop

As tarefas periódicas do sistema são definidas em:

Tarefa periódica que faz leitura da temperatura e codificação dos sinais;

Tarefa periódica que faz comparação das temperaturas para acionamento dos atuadores;

Tarefa periódica que indica o estado atual do sistema.

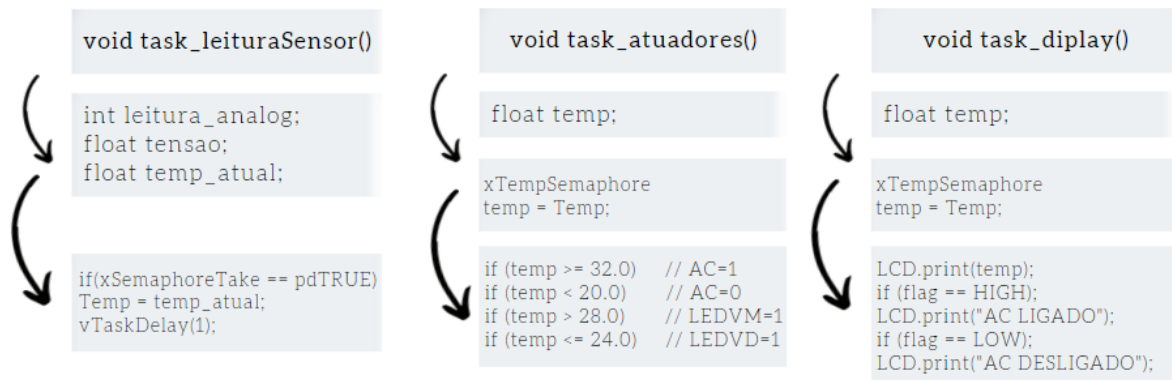


Figura 4.3 - Tarefas periódicas do sistema

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] How to use FreeRTOS structure Queue to Receive Data from Multiple Tasks. Microcontrollers Lab. Disponível em: <<https://microcontrollerslab.com/arduino-freertos-structure-queue-receive-data-multiple-resources/>>. Acesso em: 21 Aug. 2021.
- [2] BERTOLETI, Pedro. Principais conceitos de RTOS para iniciantes com Arduino e FreeRTOS. Embarcados - Sua fonte de informações sobre Sistemas Embarcados. Disponível em: <<https://www.embarcados.com.br/rtos-para-iniciantes-com-arduino-e-freertos/>>. Acesso em: 21 Aug. 2021.
- [3] Arduino - AnalogRead. Arduino.cc. Disponível em: <<https://www.arduino.cc/en/Reference/AnalogRead>>. Acesso em: 21 Aug. 2021.
- [4] SENSOR DE TEMPERATURA LM35. Sensor de Temperatura LM35. Bau Eletrônica. Disponível em: <<https://www.baudaeletronica.com.br/sensor-de-temperatura-lm35.html#:~:text=O%20Sensor%20de%20Temperatura%20LM35,cada%20grau%20celsius%20de%20temperatura.>>. Acesso em: 21 Aug. 2021.
- [5] How to “Multithread” an Arduino (Protothreading Tutorial). Arduino Project Hub. Disponível em: <<https://create.arduino.cc/projecthub/reanimationxp/how-to-multithread-an-arduino-prototyping-tutorial-dd2c37>>. Acesso em: 25 Aug. 2021.

