Projeto Air Controller

Autor: Bruno Cayres Messias

Requisitos do Sistema

Necessidade do Cliente

Redução de consumo de energia elétrica em sistemas de ar-condicionado, devido o mesmo estar ligado sem estar presente um usuário no recinto ou alguma porta ou janela aberta.

O sistema irá cortar diretamente a alimentação do sistema de ar-condicionado, com possibilidade de portabilidade para diversos fabricantes.

Requisitos

Requisitos Funcionais

- Possuir funcionalidade Admin(técnico) para análise de dados
- Sinal luminoso para aviso de aberturas no recinto
- Sistema de envio de Logs via comunicação UART e Wireless

Requisitos Não-Funcionais

- Baixo consumo de energia, abaixo de 5 mA para 5V
- Baixo custo de fabricação abaixo de R$30
- Dimensões máximas do sistema 10cmx10cmx2cm
- Encapsulamento com classe de proteção mínima IP20

Metas e Objetivos

Metas	**Objetivos**
OBJ1. Reduzir o Consumo de energia elétrica	**OBJ1.1** Desligar o ar-condicionado caso haja uma abertura.
	OBJ1.2 Impedir o acionamento do ar-condicionado caso ainda houver alguma abertura.
OBJ3. Deve possuir um sistema de sinalização para o usuário	**OBJ3.1** Deve possuir um LED ligado sempre que houver alguma abertura no recinto
OBJ.4 Deve possuir entradas e saídas de informações	**OBJ4.1** Deve existir um botão de acionamento par que o usuário possa inicializar o ar-condicionado
	OBJ4.3 Deve possuir um sistema de envio de logs via transmissão UART e Wireless

Especificação do Sistema Embarcado

Descrição do Sistema

O sistema visa implementar um controlador para o uso mais eficiente do ar-condicionado, o sistema irá desligar o ar-condicionado sempre que uma porta ou janela seja aberta e irá impedir que o sistema seja ligado caso as mesmas permaneçam abertas, para isso um sinal de LED será acendido, indicando que alguma porta ou janela ainda está aberta, quando todas as aberturas serem fechadas o LED será desligado liberando que o usuário possa ligar o ar-condicionado, no entanto, caso alguma porta ou janela seja aberta o sistema irá desligar o ar-condicionado.

Fluxograma do Sistema

A seguir um fluxograma, ilustrando o funcionamento do sistema.

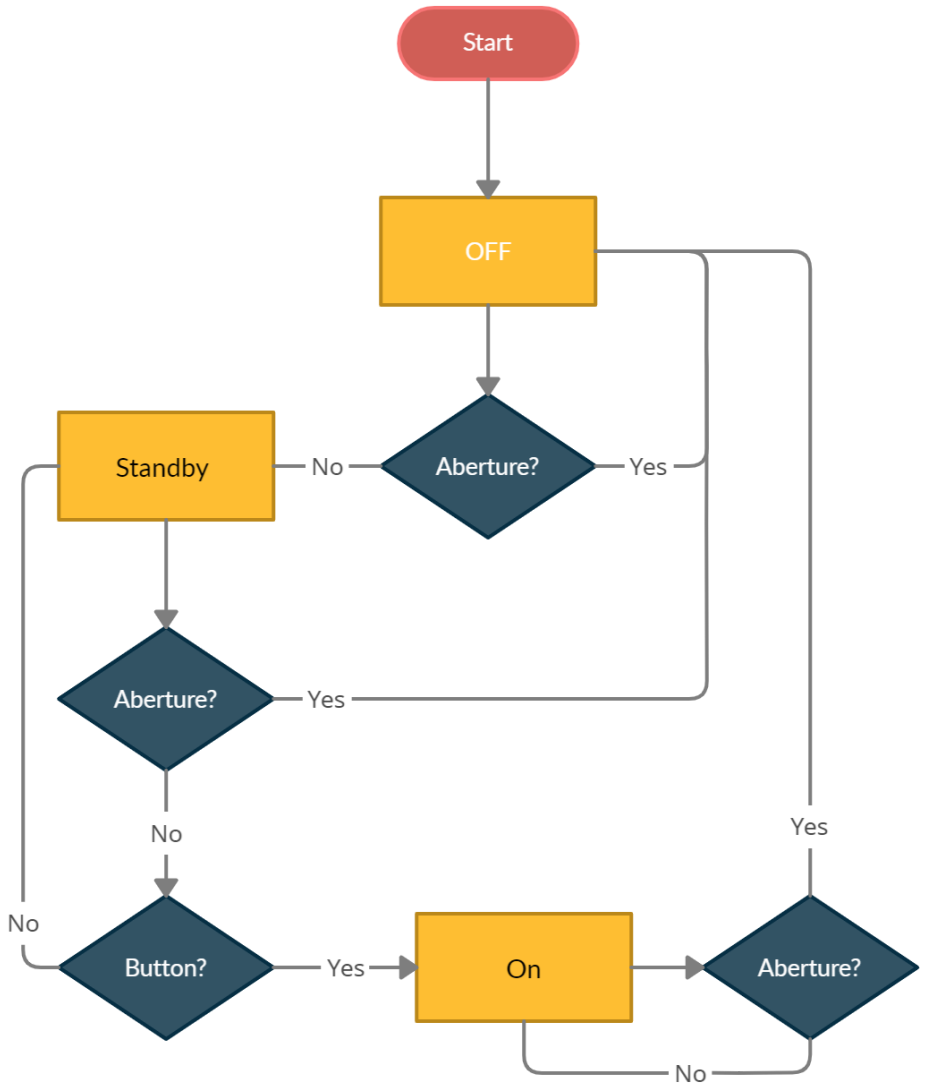
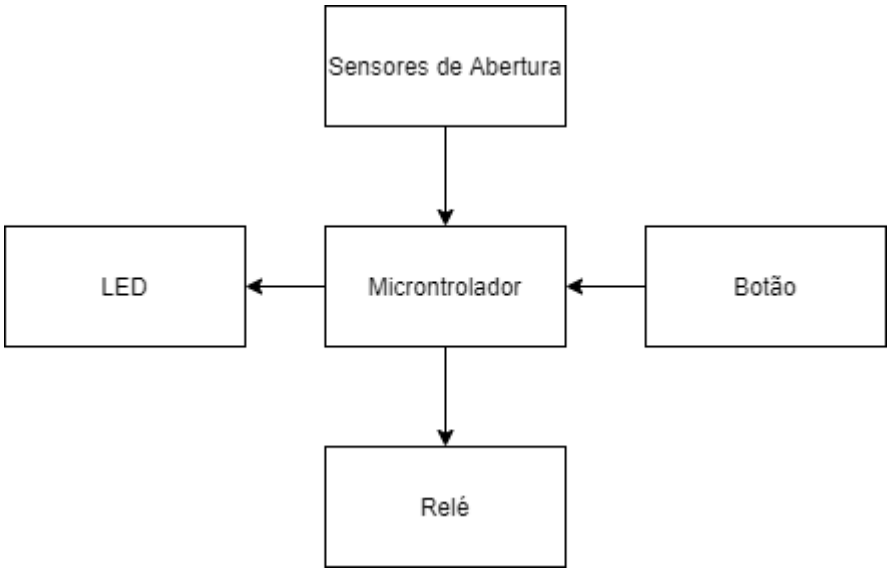


Diagrama de Blocos

A seguir o Diagrama de Blocos proposto



Arquitetura do Sistema Embarcado

Descrição da Arquitetura

Será utilizado para o sistema o microcontrolador ATmega328P, devido seu pequeno baixo consumo de energia(1.5mA em 3V para 4 MHz de clock, no modo ativo), possuindo um intervalo de operação de 2.7V até 5.5V, e possuindo um intervalo de operação na faixa de -40°C à 125°C, sendo mais que o suficiente para o pleno funcionamento do sistema.

Componentes de Software

O software será desenvolvido em C + +, sem a necessidade de um sistema operacional, utilizando conceitos de programação orientada a objetos, se utilizando de polimorfismos e classes abstratas, visando a portabilidade do código para diversos microcontroladores e sistemas de ar-condicionado. O sistema será compilado utilizando o compilador AVR-GCC, e utilizando bibliotecas AVR, disponibilizadas pela AVR-libc, e será programado pela IDE Atmel Studio, disponibilizado pela Microchip, que realizará a compilação do programa onde será gerado um arquivo .hex para posteriormente carregar no microcontrolador via Bootloader.

Componentes de Hardware

O sistema completo irá Incluir os seguintes componentes:

- Microcontrolador ATmega328P(7mmx7mmx1mm)(R$6,81)
- 1 Relé(43 mm x 17 mm x 19mm)(R$9,90)
- 1 LED(5mmx5mm)(R$0,24)
- 1 Botão de pulso(22mmx22mm)(R$12,35)

Tendo um custo total de R$29,3, abaixo do limite de R$30,00.

Máquina de Estados

A seguir a Máquina de Estados proposta:

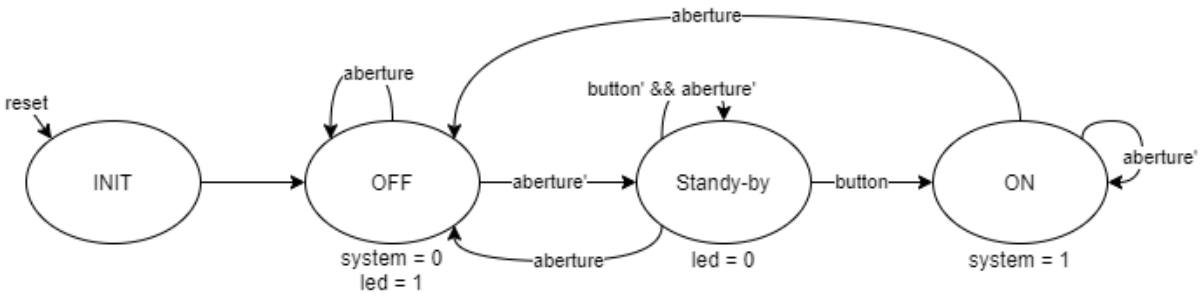
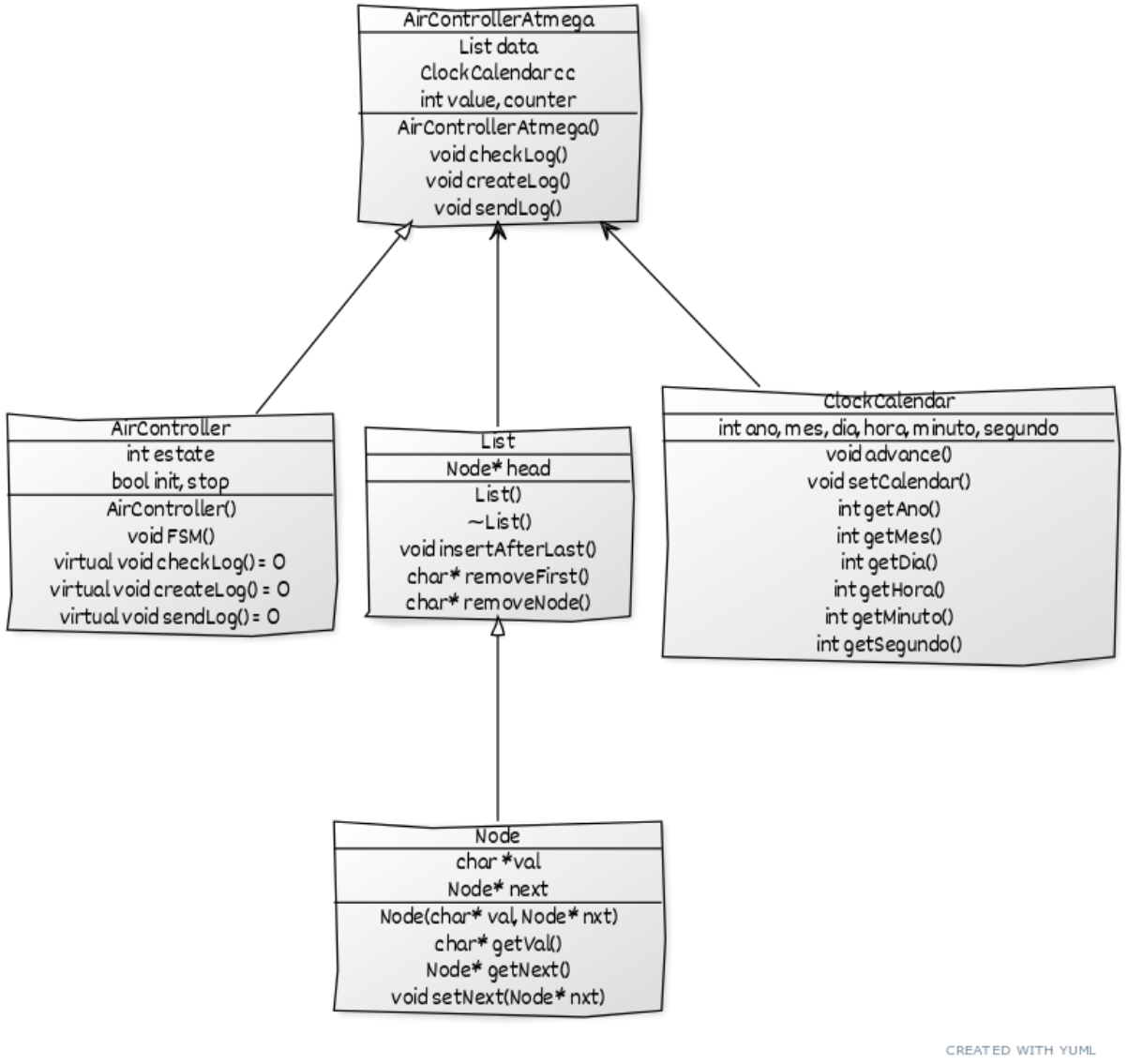


Diagrama de Classes

A seguir o diagrama de classes proposto



https://yuml.me/brunomessias/13281894.jpg

Sistema de Logs

O sistema de logs irá enviar via comunicação UART, por meio de interrupções específicas, com um Baud Rate de 9600, sem bit de paridade e um stop bit, para facilitar o debug e apresentar métricas importantes para o Administrador.

Será enviado um log contendo o ID do controlador, Data/Hora do log e qual foi o evento ocorrido, podendo ser um dos seguintes casos:

- O Sistema desligou devido uma abertura de porta ou janela(‘a’)
- O Sistema foi iniciado por um usuário(‘b’)

O envio desses dados serão enviado em uma string seguindo o seguinte modelo:
“ID/ano:mes:dia:hora:minuto:segundo/evento”

Onde o evento será identificado como ’a’ ou ’ b’ mapeados para cada evento anteriormente.
A seguir um exemplo de log enviado:
“01/21:03:30:15:05:45/a”

Onde é um log do sistema com seu Id = 1, que ocorreu no dia: 30/03/2021, às 15:05:45, onde ocorreu um desligamento do sistema devido uma abertura.

Enquanto os dados não são transmitidos, o sistema irá armazenar em uma fila de dados para posteriormente ser esvaziada ao transmitir o log, esta fila possui as operações de inserir um dado antes do último adicionado e a função de retirar o primeiro valor adicionado.

Software do Computador

Descrição do Sistema

Toda a comunicação enviada pela porta serial será lida pelo computador armazenado os logs numa fila de dados e oferecerá ao usuário funções para a visualização dos logs armazenados, entre elas estão:

- Listar todos os logs em um intervalo de tempo
- Apresentar o tempo total que o sistema manteve o ar-condicionado alimentado

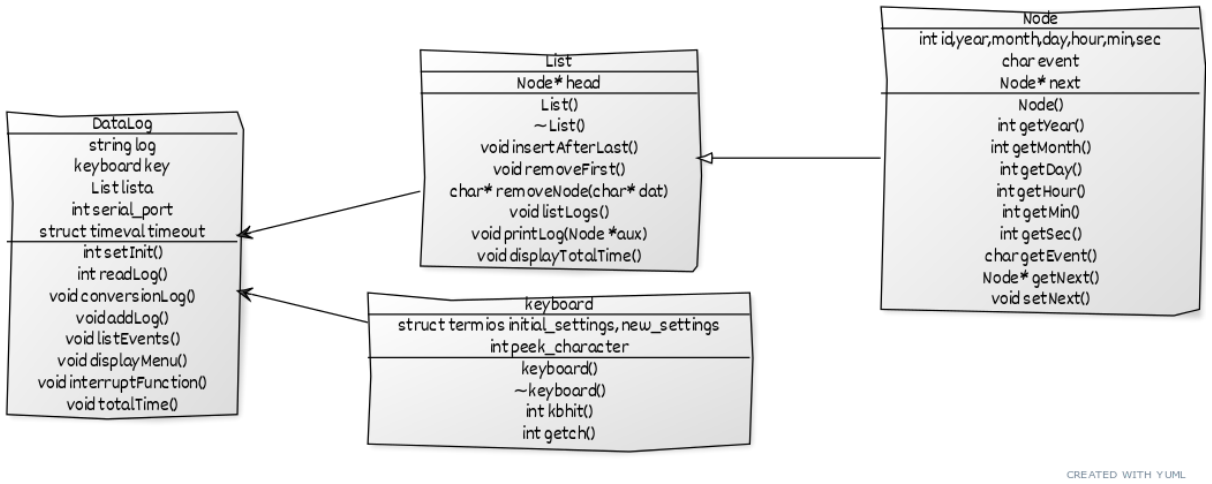
A fila de dados possui as duas operações descritas acima e a função de adicionar um dado antes do último adicionado.

Este software será implementado em software, utilizando a linguagem de programação C++, padrão 11, sendo compilado no Sistema Operacional Linux, utilizando o compilador g++. Será utilizado bibliotecas nativas do Linux para controle de arquivos e bibliotecas padrão de C++.Será compilado através de diretivas de um MakeFile específico e será executado via terminal pelo administrador. Haverá um menu para que o usuário do software consiga selecionar a apresentação dos dados armazenados.

No terminal o administrador pode executar o programa seguindo os seguintes comandos:
make computer
./computer

Diagrama de Classes

A seguir o Diagrama de classes proposto.



https://yuml.me/brunomessias/8a7c6ddc.png

Software do Smartphone

Descrição do Sistema

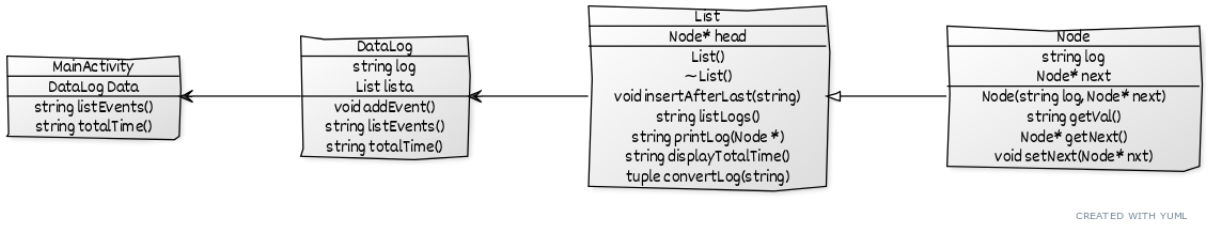
Toda a comunicação enviada pela porta serial será lida pelo smartphone armazenado os logs numa fila e oferecerá ao usuário funções para a visualização dos logs armazenados, entre elas então:

- Listar todos os logs em um intervalo de tempo;
- Apresentar o tempo total que o sistema manteve o ar-condicionado alimentado;

Será desenvolvido um aplicativo para o sistema Android, usando a plataforma Android NDK e OpenGL utilizando a IDE Android Studio, onde será compilado, disponibilizando um arquivo .apk que poderá ser executado em um smartphone Android, em uma versão Lollipop (5.0) ou superior

Diagrama de Classes

A seguir o Diagrama de classes proposto:



https://yuml.me/brunomessias/smarthphone.png

Plano de Testes

A seguir será apresentado o plano de teste para a verificabilidade dos três softwares.

Plano de Testes Embarcado

1. **Checando a Inicialização**
    a. Inicializar o sistema com o sinal de porta aberta(Esperado: sinal LED vermelho)
    b. Inicializar o sistema com o sinal de porta fechad(Esperado: nenhuma sinalização)
2. **Checando leitura das portas**
    a. Inicializar o sistema com porta fechada e observar o apagamento do LED vermelho ao inserir o sinal de porta fechada
3. **Checando a leitura do botão para ativar o sistema**
    a. Com o sistema no estado Standby(Portas fechadas), observar o acendimento do LED amarelo indicando a ativação do sistema e manter a sua sinalização
4. **Checando a leitura das portas com o sistema ativado**
    a. Com o sistema no estado ON(LED amarelo ligado), observar o acendimento do LED vermelho e o desligamento do LED amarelo após o sinal de porta fechada.
5. **Checando o envio de LOGs**
    a. **Observar com o monitor de porta serial o envio do log para os dois seguintes casos:**
        i. Sistema desligado por abertura
        ii. Botão de acionamento pressionado

Plano de Teste Software Computador

1. **Checando sinal de log armazenado na ocorrência de um log**
    a. Na ocorrência de um log enviado analisar o print no terminal “Log Stored!”
2. **Checando consultas no log**
    a. Checar o armazenamento do log na fila de dados(Enviar 4 logs (2 de acionamento e 2 de desligamento):
        i. Checar nas opções de listar os logs enviados o aparecimento destes.
        ii. Checar a correta contagem do tempo (em minutos) em que o sistema esteve ligado.

Plano de Testes Software Smartphone

1. **Checar na inicialização o aplicativo, o aparecimento de botões(LOGs) e (Cont Time)**
2. **Checar o funcionamento dos Botões**
    a. Dentro do software estará armazenado 4 logs, espaçados de 10 minutos na seguinte sequência: On -> Off -> On -> OFF
    b. Checar no botão LOGs, o aparecimento destes logs referenciados
    c. Checar a contagem correta do tempo, ao pressionar o botão Cout Time, com base nestes logs
    (Esperado: 20 min)

Desempenho

A seguir será analisado alguns desempenhos calculados a partir das ferramentas de compilação

Desempenho Software Embarcado

Dado a compilação utilizando o compilador AVR-gcc e o carregamento no microcontrolador Atmega 328P, através da ferramenta AVRdude, e a ferramenta de análise Metrix++, obtemos os seguintes parâmetros:

1. Oscilador: 28.800 kHz
2. Período de Clock para bootloader: 3.3 us
3. Tempo de Escrita no Microcontrolador: 0.6s
4. Tempo de Leitura no Microcontrolador: 0.47s
5. Tamanho do código: 3185 bytes
6. Complexidade do código, segundo Mccabe's Cyclomatic Complexity: 1.27
7. Tempo de execução máxima: 0.01146173 s

Desempenho Software Computador

A seguir a métricas obtidas para o software para o computador, através da ferramenta Metrix++:

1. Complexidade do código, segundo Mccabe's Cyclomatic Complexity: 1.69
2. Tempo de execução máxima: 0.04639673 s
3. Tamanho do código: 8342 bytes

Desempenho Software Smartphone

A seguir a métricas obtidas para o software para smartphone, através da ferramenta Metrix++:

1. Complexidade do código, segundo Mccabe's Cyclomatic Complexity: 0.76
2. Tempo de execução máxima: 0.03762221 s
3. Tamanho do código: 3652 bytes

Testes e Demonstrações

Todos os códigos foram testados e podem ser observados pelo seguinte vídeo: https://youtu.be/AJ2PtJMftwc

Para fins de clareza, o sinal de LED vermelho indica que o sistema está indicando que há alguma abertura, já o LED amarelo indica que o sistema está ativo.

Source Code

Todos os códigos podem ser encontrados no GitHub, através do link: https://github.com/Bruno-Messias/AirController