



Universidade Federal de Santa Catarina – UFSC  
Campus Universitário de Florianópolis  
Curso de Engenharia Eletrônica  
Disciplina EEL7323 – Programação C++ para Sistemas Embarcados  
Professor Dr. Eduardo Augusto Bezera

## **Controlador de ativação do Ar Condicionado**

### **Documentação I**

Aluno:      Alexandro Vanderley dos Santos    Mat.: 14201049

Florianópolis, 22 de abril de 2021.

# 1 Projeto : Controlador de ativação do Ar Condicionado

## 1.1 Especificação

O sistema será usado para evitar o gasto de energia elétrica quando alguma abertura, porta ou janela, estiver aberta. Para isso, a energia fornecida pela rede elétrica deverá ser cortada. Deverá existir um botão de acionamento que ao ser pressionado irá reestabelecer fornecimento de energia elétrica desde que os sensores não indiquem alguma porta ou janela aberta. Na Figura 1 vemos um esquema do projeto proposto.

Um led deverá acender se o sistema está fornecendo energia para o ar condicionado. O botão deverão ter livre acesso pelo usuário.

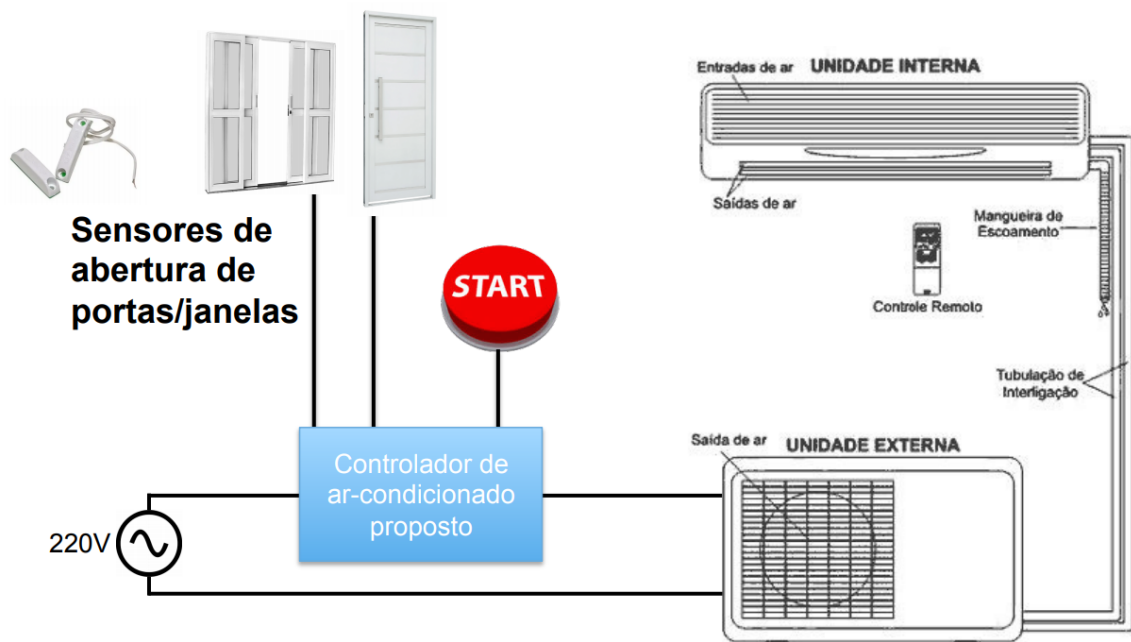


Figura 1: Esquemático do projeto proposto.

## 1.2 Lista dos requisitos funcionais

- Possibilitar que um Admin (técnico) ajuste a configuração de data e hora;
- Possuir tecla para reestabelecer energia ao ar condicionado;
- O ar condicionado precisa ser desligado caso alguma porta ou janela seja aberta;
- O ar condicionado deverá permanecer desligado até que as aberturas sejam fechadas;
- Fornecer, via serial ou WiFi, relatório de atividades.

## 1.3 Lista dos requisitos não funcionais

- O produto precisa ter um custo no máximo de R\$100,00;
- O consumo máximo deve ser de 100 mA e 150 mA com WiFi;
- Dimensões devem de no máximo 10x15x5 cm;
- Classe de proteção IP6

## 1.4 Hardware utilizado

O hardware, representado no diagrama da Figura 2, é composto por:

- Raspberry PI Pico, microcontrolador RP2040;
  - CPU: Dual-core Arm Cortex M0+ processor, flexible clock running up to 133 MHz
  - 264KB of SRAM, and 2MB of on-board Flash memory
  - Low-power sleep and dormant modes
  - 26 × multi-function GPIO pins
  - 2 × SPI, 2 × I2C, 2 × UART, 3 × 12-bit ADC, 16 × controllable PWM channels
  - 8 × Programmable I/O (PIO) state machines for custom peripheral support
- Um estágio de potência com relé, acionados por transistores e monitorado por led;
- Fonte de alimentação regulada, usando transformador 6V+6V 250mA;
- Módulo serial/WiFi Telnet;
- Módulo conversor serial/USB CH340;
- Botão de acionamento (Start);
- Entrada para dois sensores do tipo normalmente fechado.

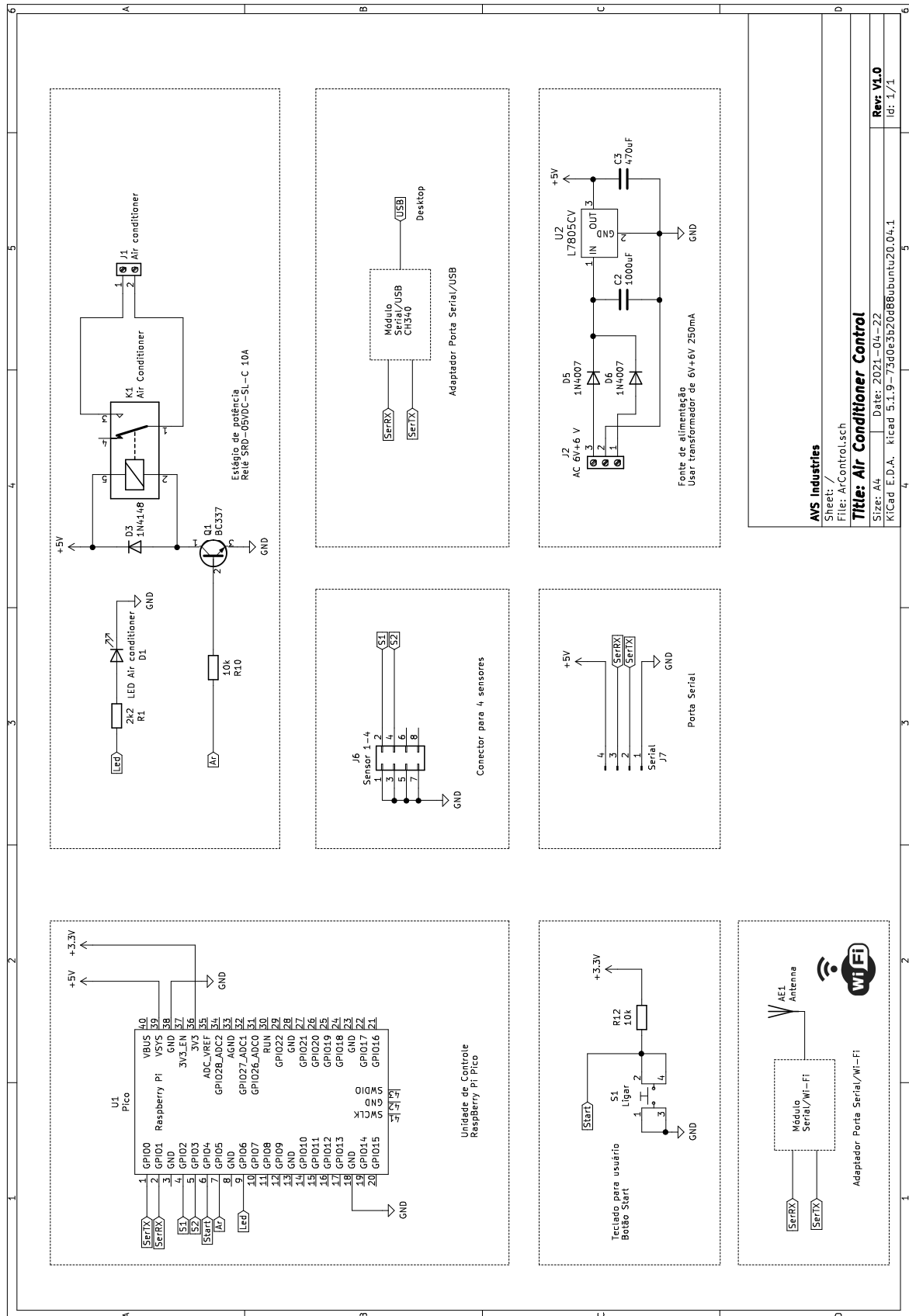


Figura 2: Esquemático

## 2 Software

### 2.1 Hardware

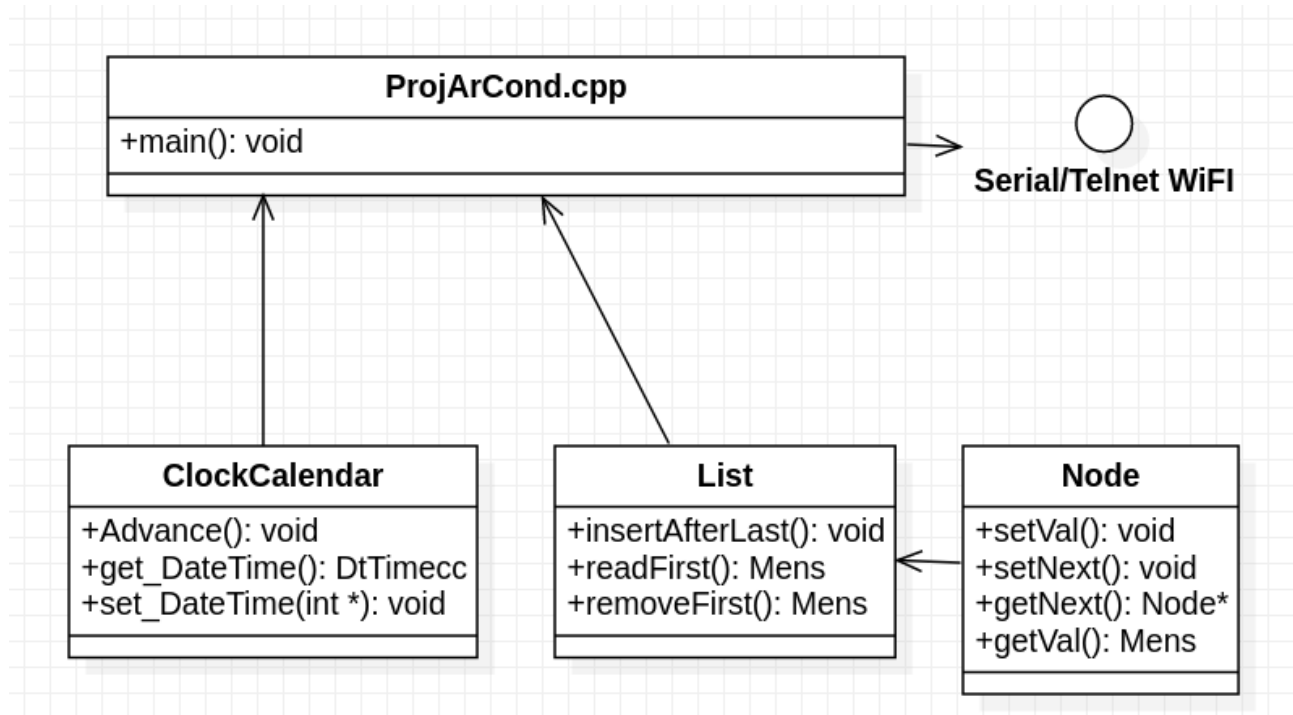


Figura 3: Diagrama de classes do software do hardware.

Utilizando o diagrama de classes da Figura 3 o programa, embutido no hardware, deve proporcionar:

- O software deverá ser capaz de interagir com um computador através da porta serial ou WiFi;
- Deverá proporcionar as opções de configuração de data e hora;
- Fornecer relatório de atividades antes armazenada na forma de lista e apagadas;
- Detectar e impedir que o relé seja acionado caso alguma porta ou janela, do ambiente, esteja aberta;
- Receber comandos via serial/WiFi para acionamento e desligamento do relé.

#### 2.1.1 Compilação

Para compilar o programa é necessário obter o SDK do Raspberry Pi Pico, conforme indicado no cabeçalho do programa principal.

## 2.2 Software Linux

O software, desenvolvido para a distribuição Linux Mint 20.1 em C++, conta com dois painéis. O painel *Controle do ar Condicionado* permite a ativação/desativação do equipamento, verificar a data e hora do hardware (Figura 4), atualizar data e hora do hardware e solicitar o relatório de utilização (Figura 5). Um segundo painel foi adicionado para mostrar informações relevantes à opção escolhida no painel anterior.

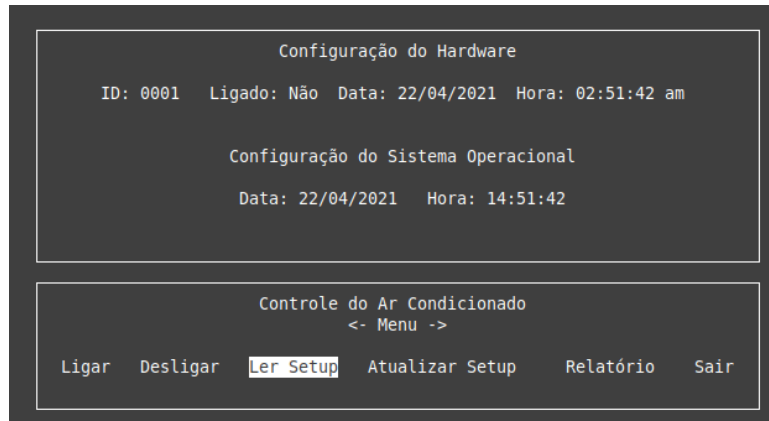


Figura 4: Tela do software Linux.

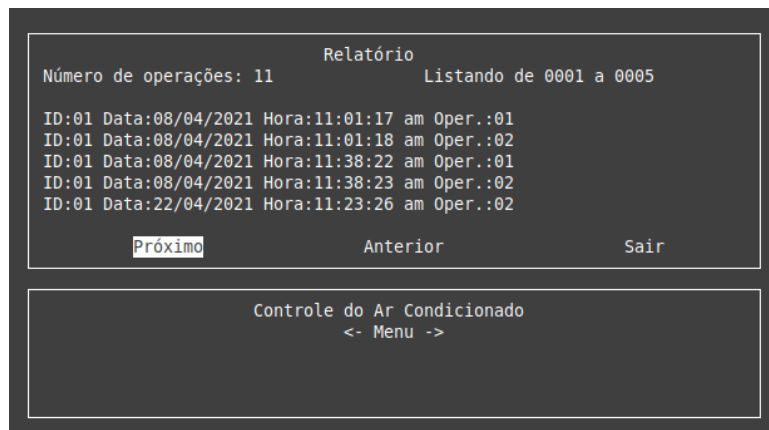


Figura 5: Tela de relatório.

Todas as operações são arquivadas no arquivo *relatorio.dat* cumulativamente, propiciando uma análise futura dos dados mais apurada.

### 2.2.1 Compilação e execução

Para compilar o programa basta executar:

```
# g++ -o ProgAr ProgAr.cpp -lnurses
```

**Obs.:** A biblioteca *nurses*, versão 5.0 ou mais recente, deve estar instalada.

Depois de compilado basta executar:

```
# ./ProgAr
```