



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

Procedimento de testes

Aluno: Alexandro Vanderley dos Santos Mat.: 14201049

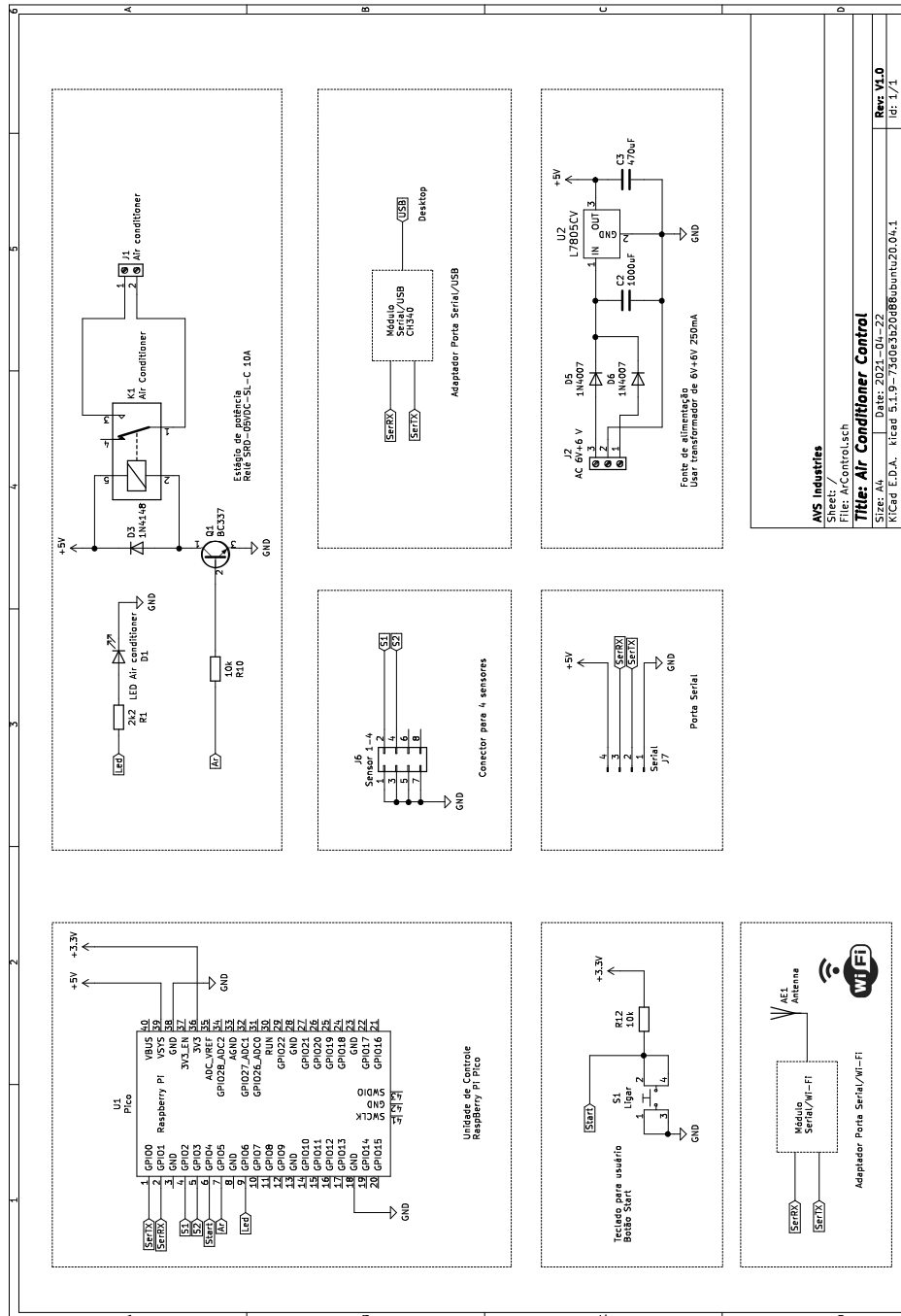
Florianópolis, 09 de maio de 2021.

1 Procedimento de testes do Controlador de ativação do Ar Condicionado

1.1 Hardware

Utilizando o circuito da Figura 1, uma série de teste realcionados à alimentação do circuito como conexão serial, acionamento indesejado, acionamento por software, prevenção de múltiplas requisições com Debounce, entre outras, a fim de validar o funcionamento do equipamento, foram realizadas.

Figura 1: Esquemático

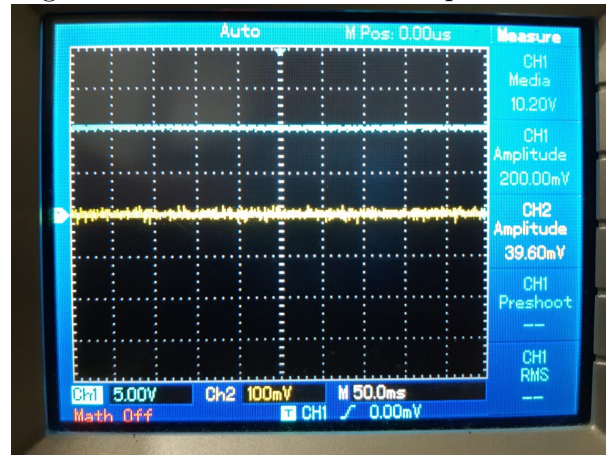


Fonte: O autor, 2021.

1.1.1 Fonte de Alimentação

Utilizando um transformador isolador de 7,5V+7,5V, com 250 mA de capacidade de corrente, a parte de retificação e estabilização apresentou, sobre C2, uma tensão de 10,4 V com ripple de 39,6 mV, significando uma variação de 0,386% , conforme figura 2.

Figura 2: Medida com osciloscópio sobre C2.



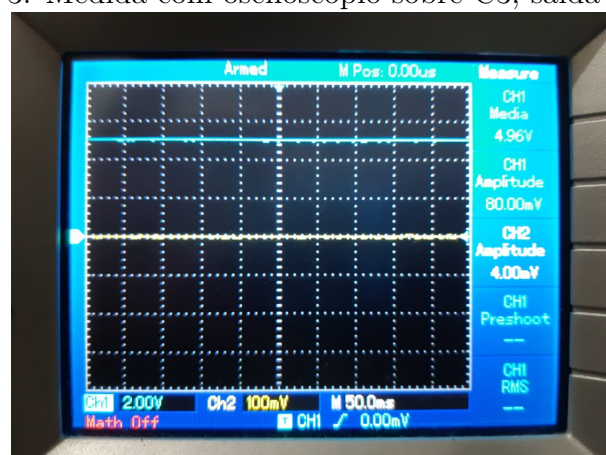
Fonte: O autor, 2021.

Na saída da fonte, sobre C3, a tensão ficou em 4,96 V e um ripple de 4 mV, Figura 3. Esta tensão está em um nível aceitável, pois as tensões limites para o funcionamento do circuito, sem risco de queima ou desligamento, é de 4,8 V, como mínimo, e de 5,25 V, como máximo. Também dentro do aceitável, o ripple, está abaixo de 5%. Caso este valor viesse a ser ultrapassado ocasionaria instabilidade no funcionamento. Como condição para este teste, o circuito estava com carga máxima, ou seja, circuitos, leds e rele acionados.

1.1.2 Estágio de potência

No estágio de potência, testes de acionamento foram realizados. Como o módulo de processamento, o Raspberry PI Pico, emite um sinal de aproximadamente 3,12 V em seus terminais, podendo variar entre 3V e 3,18 V. A utilização do transistor Q1, o BC337, com um resistor de polarização direta de 10 k Ω , garantiram o acionamento do relé. Apesar de Q1 possuir uma tensão $V_{CE} = 0,3$ V, com uma corrente $I_C = 1,3$ mA, o relé comportou-se adequadamente.

Figura 3: Medida com osciloscópio sobre C3, saída da fonte.

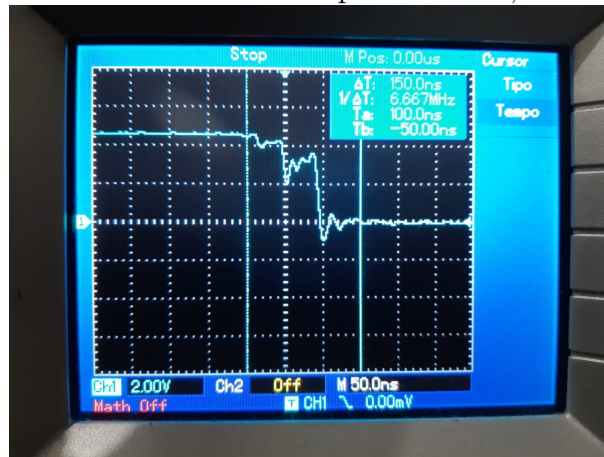


Fonte: O autor, 2021.

1.1.3 Botão Start

Como esperado para um botão de pressão mecânico, ruídos durante o acionamento foram detectados. Para solucionar este tipo de problema, no software do Raspberry, um atraso na operação do circuito, neste projeto de 150 ms, foi adicionado. Assim que o botão é pressionado, o programa deve aguardar 150 ms para que o tempo de transição do estado da porta estabilize e não aconteçam acionamentos múltiplos, embora este processo não gere coleta de informações extras ou qualquer operação no relé, pois outra forma de prevenção foi utilizada pelo software. O tempo de transição da porta foi de aproximadamente 150 ns, Figura 4, 1000 vezes menor que o tempo utilizado pelo programa.

Figura 4: Medida com osciloscópio sobre C3, saída da fonte.



Fonte: O autor, 2021.

1.1.4 Módulo de processamento

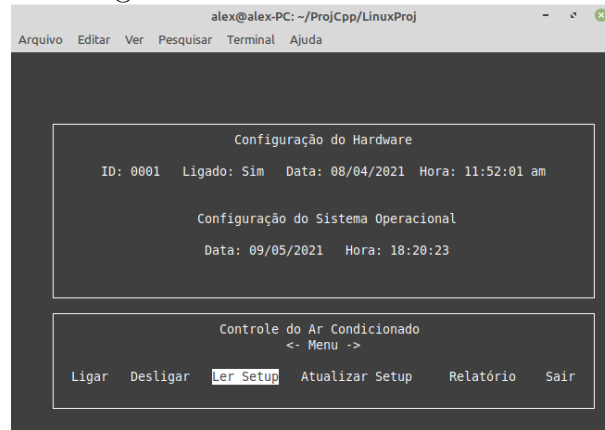
Como módulo de processamento, foi utilizado o Raspberry Pi Pico. Este pode ser alimentado com tensão muito inferior ao necessário pelos outros componentes, então, neste quesito, apenas deve-se garantir que a tensão máxima não ultrapasse os 5,25 V, o mesmo deve ser considerado para o módulo WiFi¹, e isto foi garantido pelo regulador de tensão na fonte LM7805, U2.

¹O módulo WIFI, não pôde ser testado, este foi relacionado na lista de componente para fins de requisito de projeto. Para seção de teste de software, o referido não será citado por não ser possível realizar os testes adequados.

1.2 Software

O aplicativo desenvolvido para o S.O. Linux, Figura 5, conta com a função de integrar, através da porta serial, o hardware ao computador. O aplicativo será capaz de comandar e requisitar relatórios de utilização.

Figura 5: Tela do Software Linux.



Fonte: O autor, 2021.

1.2.1 Acionamento ou desligamento do PC

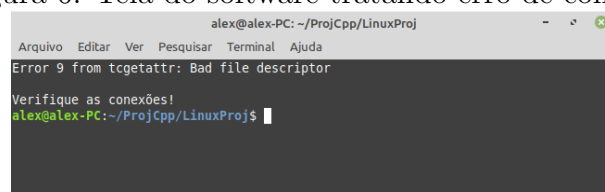
O hardware, tem total autonomia para o funcionamento, não necessitando de um computador para sua utilização. Mas, quando o hardware é conectado ou desconectado ao computador, ou mesmo o computador é ligado ou desligado, podem ocorrer falhas e realizações inesperadas de operações por parte do circuito desenvolvido. No entanto, para que este tipo de erro seja evitado, no software do Raspberry, foi inserido uma sequência de comandos que, apenas com o recebimento de comandos completos e corretos, alguma operação é realizada, mesmo que o desligamento repentino do pc venha a ocorrer. Variações repentinas de tensão na porta serial são absorvidas pelo módulo conversor USB/Serial.

1.2.2 Porta Serial

A utilização da porta serial, para este projeto está limitada para /ttyUSB0 e a /ttyUSB1. Realizando conexões e desconexões repetitidas vezes, em um curto tempo, ocasionou uma falha na detecção do dispositivo. Isto deve-se ao fato de que o S.O. Linux necessita de um período de atualização da condição das portas seriais que, variam conforme o modelo e especificações do hardware utilizado na confecção do pc.

Realizando o teste diversas vezes, notou-se que em algum momento a porta destinada ao dispositivo não correspondia às delimitadas, gerando um erro que foi tratado com êxito pelo programa, pois o mesmo veio a ser interrompido, conforme programado, Figura 6. Como já citado o dispositivo não realizou nenhuma operação durante este processo.

Figura 6: Tela do software tratando erro de conexão.



Fonte: O autor, 2021.

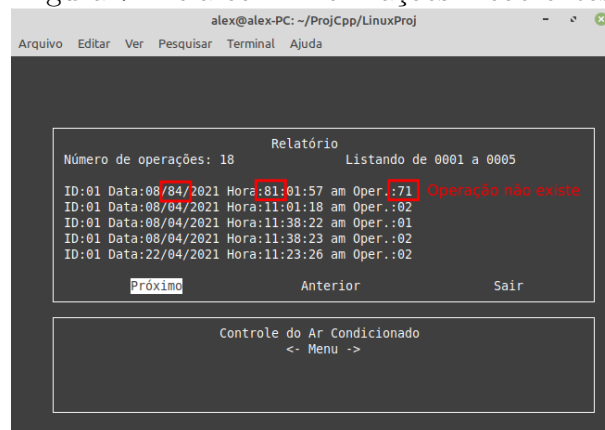
1.2.3 Acionamento/Desligamento do ar condicionado via software

Para este teste, foram realizadas diversas operações de acionamento e desligamento, intercalando com outras operações, como obtenção de relatórios e configuração do relógio do dispositivo. Com as diversas ferramentas de controle de erro, tanto no dispositivo, quanto no aplicativo Linux, mesmo executando desconexões repentinas, o sistema comportou-se de forma concisa, visto que, na recepção de dados, apenas com a confirmação do pc, o dispositivo eliminaria da lista a operação. O teste envolvendo a emissão de comando incorretos ao dispositivo, como esperado, não gerou qualquer tipo de erro.

1.2.4 Obtenção de relatórios

Neste quesito, testes, durante a operação normal, não resultaram erros. Quando desconectado o dispositivo, durante a operação de recebimento dos dados, como a quantidade de dados recebida, não deve estar dentro do esperado, o mesmo não é armazenado em arquivo. Simulando uma conexão ruidosa, inserindo dados aleatórios mas com a dimensão correta, erros na apresentação do relatório, como a presença de caracteres estranhos na tela ou informação não coerente, Figura 7, são apresentados. Uma solução ainda não implementada seria a validação de cada parte da informação, verificando a coerência antes do arquivamento.

Figura 7: Tela com informações incoerentes.

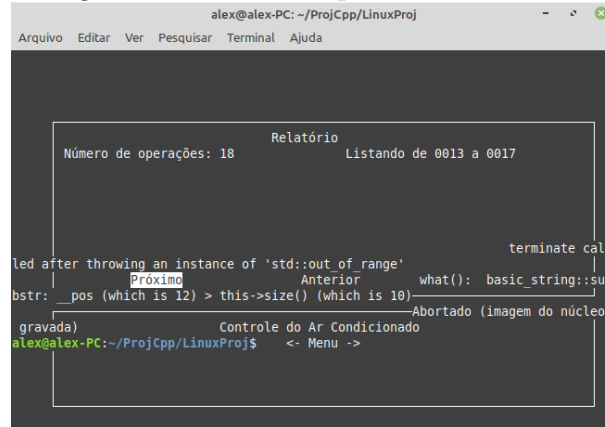


Fonte: O autor, 2021.

1.2.5 Armazenagem dos dados

Todas as informações pertinentes ao relatório são armazenadas no arquivo *relatório.dat*. Corrompendo os dados de forma proposital, erros no momento da apresentação do item corrompido geraram a exceção mostrada na Figura 8.

Figura 8: Erro no arquivo relatório.dat.



```
alex@alex-PC: ~/ProjCpp/LinuxProj
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda

Relatório
Número de operações: 18      Listando de 0013 a 0017

terminate called after throwing an instance of 'std::out_of_range':
    _pos (which is 12) > this->size() (which is 10)
Abortado (imagem do núcleo gravada)
alex@alex-PC:~/ProjCpp/LinuxProj$
```

Fonte: O autor, 2021.

Este tipo de erro pode ser evitado com a análise dos dados antes da gravação, como mencionado no item anterior, ou reutilizar a função de validação dos dados recebidos pela serial, para garantir que o mesmo não esteja corrompido. Quanto ao problema físico de disco, por exemplo, uma verificação inicial de todo arquivo pode informar tal acontecimento e, possivelmente, recuperar parte da informação não danificada.

2 Conclusão

O testes realizados se mostraram capazes de indicar vulnerabilidades e ponto de melhoramento de todo conjunto, hardware e software. Embora todo trabalho feito tenha um resultado positivo, problemas envolvendo situações onde o desenvolvedor não possui controle, como o defeito em disco ou ruídos na comunicação, precisam ser incluídos como requisito de segurança para garantir um melhor aproveitamentos do sistema, sem falhas ou interrupções.