

DAS Departamento de Automação e Sistemas
CTC **Centro Tecnológico**
UFSC Universidade Federal de Santa Catarina

Desenvolvimento de um Sistema de Acionamento Para Equipamentos de Soldagem

Relatório submetido à Universidade Federal de Santa Catarina

Como requisito para a aprovação na disciplina

DAS5511:Projeto de Fim de Curso

Ariel Bruscatto

Florianópolis, 12 de Agosto de 2016

Desenvolvimento de um Sistema de Acionamento Para Equipamentos de Soldagem

Ariel Bruscatto

Esta monografia foi julgada no contexto da disciplina
DAS5511:Projeto de Fim de Curso
E aprovada na sua forma final pelo
Curso de Engenharia de Controle e Automação

Prof. Werner Kraus Junior

Assinatura do Orientador

Banca Examinadora:

Régis Henrique Gonçalves e Silva
Orientador na Empresa

Prof. Werner Kraus Junior
Orientador no Curso

Prof. Marcelo Ricardo Stemmer
Avaliador

Guilherme Keiji Saito
Eduardo Luiz Santos da Silva
Debatedores

Agradecimentos

Agradeço ao professor Werner Kraus Junior pela orientação e pelo incentivo.

Agradeço ao professor Régis Henrique Gonçalves e Silva pela oportunidade que me deu para desenvolver este trabalho no LABSOLDA.

Agradeço aos colegas do LABSOLDA pela amizade e pela ajuda que me deram durante os trabalhos.

Resumo

Este relatório descreve o processo de desenvolvimento de um sistema de acionamento para equipamentos de soldagem. Esse desenvolvimento foi realizado no Instituto de Soldagem e Mecatrônica (LABSOLDA-UFSC), que está envolvido com ensino, pesquisa e desenvolvimento em Tecnologia de Soldagem (processos, procedimentos, equipamentos e Instrumentação).

O LABSOLDA possui diversos equipamentos usados nos processos de soldagem. Alguns deles, apesar de terem funções diferentes, possuem uma constituição eletromecânica semelhante. Por exemplo, há equipamentos cuja eletromecânica é composta unicamente de um motor de passo (Oscilador, ADP, Castanha, Tartilope V1). Esses equipamentos, apesar de sua semelhança, vêm sendo acionados de formas diferentes, com equipamentos diferentes.

Com o objetivo de facilitar a operação e a manutenção, foi decidido desenvolver um mesmo sistema de acionamento para esses diferentes equipamentos. Esse novo sistema é o foco deste trabalho.

O sistema em questão é composto de um hardware e um software. O Hardware é uma placa de circuito impresso (PCI) que foi desenvolvida a partir do esquemático do circuito já projetado, e tem a função de intermediar a comunicação entre um microcontrolador e os equipamentos de soldagem. Os detalhes do projeto da PCI serão mostrados neste relatório.

Do ponto de vista de software, serão discutidas neste relatório as alternativas de sua organização, considerando a possibilidade de execução sequencial ou concorrente, bem como uma comparação entre essas alternativas.

Abstract

This report describes the process of development of a drive system for welding equipment. This development was made at Instituto de Soldagem e Mecatrônica (LABSOLDA-UFSC), which is involved with teaching, research and development in Welding Technology.

LABSOLDA has many equipments used in the process of welding. Some of them, although they have different functions, have a similar electromechanical constitution. For example, there are electromechanical devices which are almost solely composed of a stepping motor (Oscilador, ADP, Castanha, Tartilope V1). These devices, despite their similarity, are being driven in different ways, with different equipment.

In order to facilitate the operation and maintenance, it was decided to develop a single drive system for these different equipment. This new system is the focus of this report.

The system is composed of a hardware and a software. The hardware is a printed circuit board (PCB) which was developed from the circuit schematic already designed, and has the function of mediating communication between a microcontroller and the welding equipment. The details of the PCB design will be shown in this report.

From a software perspective, organizational alternatives will be described in this report, considering the possibility of sequential or concurrent execution, and a comparison between these alternatives.

Sumário

Agradecimentos.....	5
Resumo	6
Abstract	7
Sumário	8
Simbologia.....	10
Capítulo 1: Introdução	11
1.1: Instituto de Soldagem e Mecatrônica	14
Capítulo 2: Placa de Circuito Impresso.....	15
2.1: Análise do Circuito	15
2.1.1: Descrição Básica dos Blocos do Circuito	16
2.1.2: Lista de Materiais	21
2.1.3: Teste de Blocos do Circuito.....	22
2.2: Fluxo de Desenvolvimento da PCI.....	23
2.2.1: Desenvolvimento de Bibliotecas.....	24
2.2.2: Transferência Lógica de Dados.....	30
2.2.3: Preparação do Layout	31
2.2.4: Desenvolvimento do Layout	36
2.2.5: Finalização do Layout.....	41
2.2.6: Arquivos de Fabricação	43
Capítulo 3: Discussões Sobre Organização do Software	44
3.1: Abordagem Sequencial Utilizando Interrupções	44
3.2: Abordagem Concorrente Utilizando um Sistema Operacional	46

Capítulo 4: Discussão dos Resultados	50
Capítulo 5: Conclusões e Perspectivas	52
Bibliografia:.....	53

Simbologia

PCI	-----	Placa de Circuito Impresso
ADP	-----	Alimentador de Pó
LED	-----	Light Emitting Diode
PTH	-----	Pin Through Hole
SMD	-----	Surface Mount Device
A/D	-----	Analógico/Digital
CAD	-----	Computer-Aided Design
IHM	-----	Interface Homem-Máquina

Capítulo 1: Introdução

Neste relatório descreve-se o processo de desenvolvimento de um sistema de acionamento para equipamentos de soldagem. Esses equipamentos são usados como auxílio para diversos processos de soldagem.

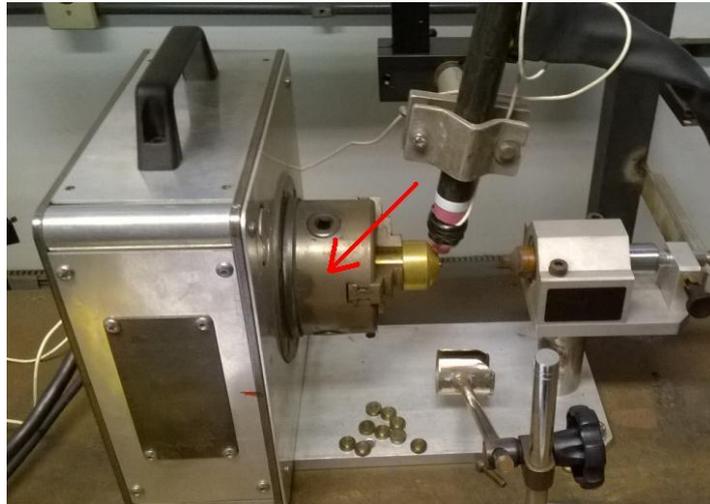
Por exemplo, o processo de soldagem ou o revestimento de tubos exige que haja um mecanismo para girar o tubo durante a soldagem. Esse giro pode ser feito por um equipamento denominado Castanha (Figura 1). A Castanha é um elemento mecânico de fixação que gira sobre um eixo. Esse processo de soldagem ou revestimento de tubos usando uma castanha consiste em fixar o tubo na castanha para que ele possa girar durante o processo de soldagem. Naturalmente, a Castanha deve funcionar em sincronia com uma tocha de soldagem.

Já o processo de soldagem utilizando alimentação de pó exige um equipamento para executar essa função, e esse equipamento pode ser um ADP (Figura 2). Um ADP pode ser usado em processos de manufatura aditiva como por exemplo o processo plasma com adição de pó. Esse dispositivo consiste basicamente em um reservatório de pó metálico e um sistema de transporte desse pó até o local onde será depositado.

Além disso, durante a soldagem pode haver a necessidade de movimentar a tocha de soldagem de maneira uniforme por uma trajetória definida. Essa movimentação uniforme pode ser realizada por um equipamento denominado Tartilope V1 (Figura 3), desenvolvido pelo LABSOLDA. Esse equipamento é um robô cartesiano capaz de fixar e movimentar uma tocha durante o processo de soldagem.

Também existem processos de soldagem em que a tocha deve oscilar com frequência e amplitude definidas. Essa oscilação pode ser implementada utilizando um equipamento denominado Oscilador Mecânico (Figura 4). O movimento de oscilação da tocha durante o processo de soldagem denomina-se tecimento e permite o aumento da área de soldagem durante a movimentação da tocha.

Figura 1 – Castanha de fixação (apontada pela seta) de peças a soldar ou revestir.



Fonte: LABSOLDA

Figura 2 –ADP (Alimentador de Pó). Na parte superior há um reservatório e na parte inferior um sistema de transporte do pó metálico.



Fonte: LABSOLDA

Figura 3 – Tartílope V1. Robô cartesiano, com IHM e fonte de soldagem.



Fonte: LABSOLDA

Figura 4 – Oscilador Mecânico. Na parte interior direita do quadrado vermelho há um motor responsável por movimentar o oscilador.



Fonte: LABSOLDA

Uma característica comum desses equipamentos é que eles são eletromecanicamente compostos de um único motor de passo. Apesar dessa semelhança, eles tem sido até então acionados através de equipamentos diferentes. Foi observado que o desenvolvimento de um sistema único de acionamento, que possa ser configurado para acionar qualquer um desses equipamentos, é uma

alternativa interessante para reduzir custos e facilitar a manutenção, a operação e a documentação. O objetivo geral deste trabalho é desenvolver esse sistema de acionamento.

Esse sistema é composto de um hardware e um software e seu desenvolvimento requer o trabalho de uma equipe multidisciplinar de pessoas, trabalhando em paralelo em partes diferentes do sistema. Este trabalho tem como objetivo específico apresentar o desenvolvimento da placa de circuito impresso (PCI) e discutir as alternativas de organização do software, considerando as possibilidades de execução sequencial e concorrente e comparando essas alternativas.

No Capítulo 2 apresenta-se o detalhamento do projeto de hardware (PCI). No Capítulo 3 serão feitas discussões sobre as possíveis abordagens de organização do software. No Capítulo 4 serão discutidos os resultados obtidos no decorrer do desenvolvimento e, no Capítulo 5 serão colocadas as conclusões e perspectivas.

1.1: Instituto de Soldagem e Mecatrônica

O Instituto de Soldagem e Mecatrônica (LABSOLDA) localiza-se no Centro Tecnológico da Universidade Federal de Santa Catarina (UFSC). Possui uma infraestrutura completa para estudo e desenvolvimento de processos e inovações na área de soldagem, sendo referência nacional nessa área de atuação.

Seus integrantes formam uma equipe multidisciplinar de mais de 50 estudantes e profissionais, entre eles estudantes de graduação e pós-graduação.

O LABSOLDA já gerou dezenas de produtos, incluindo Fontes de Soldagem, Periféricos de Soldagem, Instrumentação, Softwares, Processos e Manuais. Atualmente está realizando pelo menos 19 projetos em parceria com empresas como PETROBRAS e TRACTEBEL ENERGIA S/A. Publicou dezenas de teses e dissertações e quase 200 artigos. [2]

Capítulo 2: Placa de Circuito Impresso

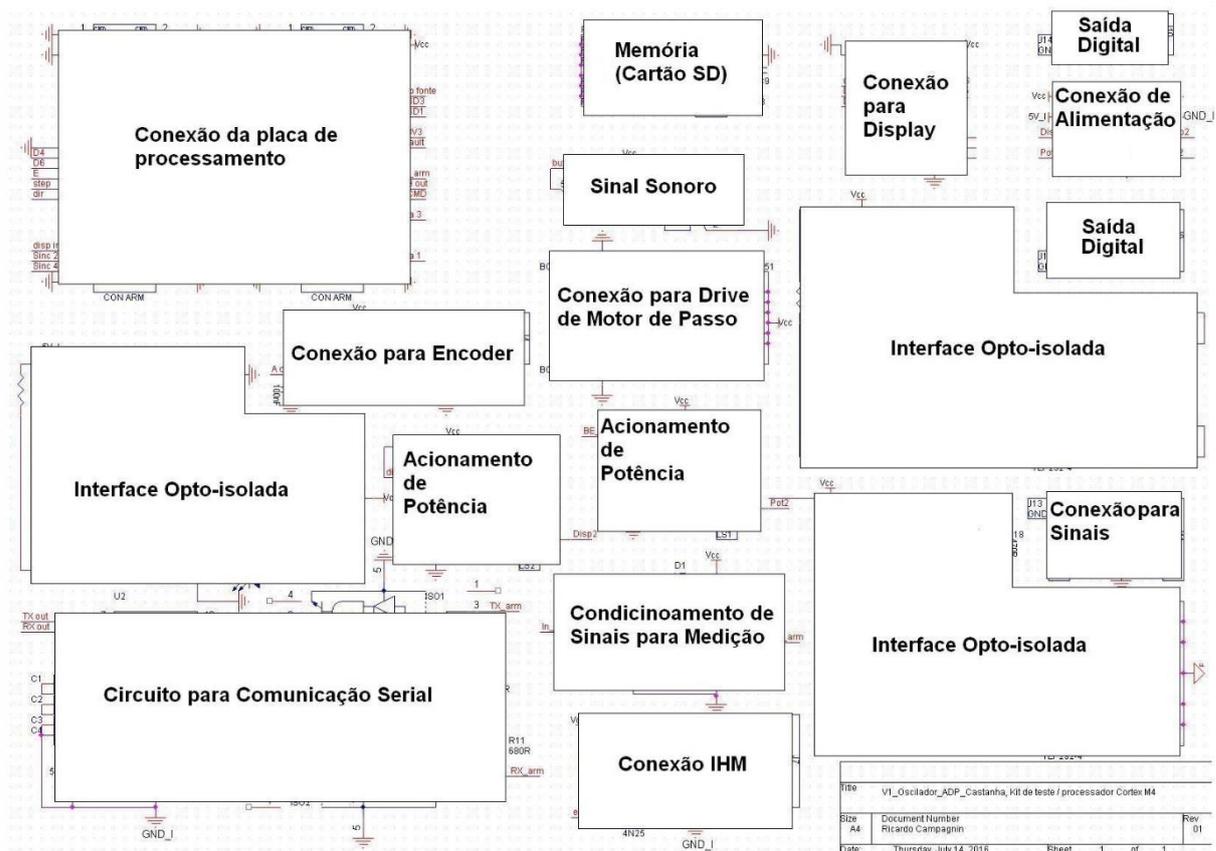
Os equipamentos de soldagem descritos na Introdução tem sido acionados por sistemas diferentes, com hardwares diferentes. Foi necessário então projetar um sistema único de acionamento, com o objetivo de reduzir custos, facilitar a manutenção e a documentação. O Hardware desse novo sistema consiste essencialmente de uma PCI, e seu desenvolvimento será descrito a seguir.

2.1: Descrição Básica do Circuito

Como pode ser observado na Figura 5, o circuito elétrico pode ser dividido em circuitos menores para facilitar a organização e o entendimento. Neste relatório os circuitos menores são considerados caixas-pretas e seus conteúdos internos, compostos de componentes eletrônicos e suas respectivas conexões, não serão detalhados, uma vez que o projeto é de propriedade do LABSOLDA (TesteSite).

O que será feito a seguir é uma breve descrição funcional de cada bloco.

Figura 5 - Diagrama de blocos do circuito.



2.1.1: Descrição Básica dos Blocos do Circuito

2.1.1.1: Conexão da Placa de Processamento

Uma vez que a placa a ser desenvolvida fará a intermediação da comunicação entre um equipamento de soldagem e um placa de processamento, é natural que haja um conjunto de conectores no circuito da Figura 5 que permita a conexão dessa placa de processamento. Esse conjunto de conectores está localizado no bloco Conexão da Placa de Processamento.

Na Figura 6 a seguir está mostrada a placa de processamento utilizada, e é possível observar nas laterais dessa placa as duas barras de pinos que deverão ser conectadas na placa a ser projetada. Essa placa de processamento foi adquirida previamente e foi referência para o projeto do circuito.

O microcontrolador da placa de processamento é o STM32F4, que possui um processador ARM de 32 bits. Esse microcontrolador é capaz de gerenciar 16 níveis de prioridades de interrupções e 16 interrupções.

Figura 6 – Placa de processamento STM32F4DISCOVERY.



Fonte: www.st.com (2016)

2.1.1.2: Interfaces Opto-Isoladas

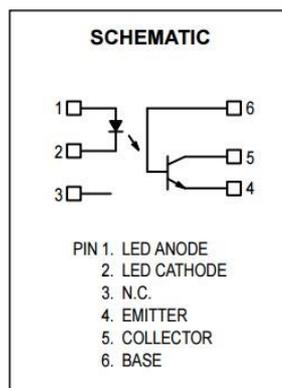
No Diagrama de Blocos do Circuito (Figura 5) há três blocos de interfaces opto-isoladas. Dois deles são responsáveis pelo isolamento de sinais digitais de saída, ou seja, sinais que saem da Placa de Processamento (Figura 6) para o equipamento de soldagem, e o último é responsável pelo isolamento de sinais de entrada provenientes da IHM, ou seja, sinais que o operador fornece para a Placa de Processamento.

O isolamento é necessário porque os dois extremos da comunicação (Placa de Processamento e Ambiente de Soldagem) funcionam com níveis de energia muito diferentes. Por exemplo, o processamento feito na Placa de Processamento envolve sinais elétricos de baixa tensão elétrica e baixa corrente elétrica, ou seja,

baixa potência elétrica. Assim, os componentes eletrônicos utilizados para o processamento são dimensionados para baixa potência elétrica. Já o Ambiente de Soldagem possui sinais elétricos de alta tensão elétrica e alta corrente elétrica, ou seja, alta potência elétrica. A razão entre os níveis de potência nos extremos da comunicação pode atingir facilmente a cada dos milhares, e ruídos elétricos que são aceitáveis em um extremo, podem danificar o outro extremo. Daí a necessidade de isolamento.

O prefixo Opto nesse contexto refere-se à luz, porque é a luz que envia informações de um extremo a outro da comunicação, por meio de dispositivos denominados opto-acopladores ou opto-isoladores. A Figura 7 mostra uma representação dessa classe de dispositivos.

Figura 7 – Representação elétrica de um opto-acoplador.



Fonte: www.datasheetcatalog.com

De um lado há um LED e do outro lado há um transistor sensível à luz. A informação percorre o sentido LED – transistor. Caso haja um ruído elétrico de alta potência no circuito de acionamento do LED, mesmo que este seja danificado, o transistor não será danificado e não transmitirá esse ruído adiante.

2.1.1.3: Circuito para Comunicação Serial

Esse circuito permite o uso do padrão de camada física RS-232 para comunicação serial entre a Placa de Processamento e o Ambiente de Soldagem. Além disso realiza o isolamento dos dois extremos da comunicação utilizando opto-acopladores.

2.1.1.4: Condicionamento de Sinais para Medição

O sistema a ser projetado deverá permitir a medição da amplitude de um sinal analógico de tensão elétrica proveniente do ambiente de soldagem. Essa medição ocorrerá de forma digital na Placa de Processamento e para isso será necessário converter o sinal analógico para um sinal digital. Antes da conversão será necessário condicionar o sinal analógico de entrada. Esse condicionamento é implementado através de um circuito de proteção que impede que esse sinal ultrapasse os níveis elétricos aceitáveis do conversor A/D, que se encontra na Placa de Processamento.

2.1.1.5: Acionamento de Potência

No Diagrama de Blocos do Circuito (Figura 5) estão representados dois blocos de Acionamento de Potência. Cada um desses blocos é um circuito que permite que um sinal de baixa potência proveniente da Placa de Processamento faça o chaveamento (ligar/desligar) de sinais elétricos de mais alta potência do Ambiente de Soldagem. Esse circuito permite também o isolamento adequado entre os dois extremos.

2.1.1.6: Sinal Sonoro

Há também um bloco indicado como Sinal Sonoro, e nada mais é que um dispositivo que emite um som com frequência definida e é acionado por um circuito conectado à Placa de Processamento.

2.1.1.7: Conexão para Drive de Motor de Passo

Esse bloco faz a conexão de sinais de acionamento vindos da Placa de Processamento a um drive externo para motor de passo. Esses sinais são utilizados para informar o número de passos a serem dados pelo eixo do motor e o sentido de rotação do eixo.

2.1.1.8: Conexão para Encoder

Esse bloco realiza o condicionamento de sinais provenientes de um encoder de quadratura e faz a conexão desses sinais à Placa de Processamento. A utilidade de um encoder nesse projeto pode ser a de permitir ao operador selecionar opções de configuração em um menu mostrado em um display.

2.1.1.9: Conexão para Display

Esse bloco faz a conexão de um display à Placa de Processamento e possui um circuito de ajuste do brilho do display.

2.1.1.10: Memória (Cartão SD)

Esse bloco faz a conexão de um cartão de memória à Placa de Processamento. Um cartão de memória pode ser usado para armazenar informações, como por exemplo informações de configuração para diferentes processos de soldagem.

2.1.1.11: Saída Digital

Há dois blocos de saída digital que disponibilizam comunicação com dispositivos externos para tarefas de sincronismo ou mesmo futuras expansões.

2.1.1.12: Conexão para Sinais

Esse bloco faz a conexão de sinais de entrada ao bloco de Condicionamento de Sinais para Medição, bem como a conexão de sinais de comunicação serial ao bloco Circuito para Comunicação Serial.

2.1.1.13: Conexão de Alimentação

Esse bloco proporciona a conexão dos sinais de alimentação provenientes de uma fonte externa.

2.1.2: Lista de Materiais

Antes de um circuito elétrico ser implementado conforme seu esquemático, é preciso que seja feita a correspondência de cada componente representado no esquemático ao seu respectivo componente real. A lista de materiais faz essa correspondência (Tabela 1).

No caso deste trabalho o formato da lista foi uma tabela. Na segunda coluna estão as referências dos componentes representados no esquemático do circuito. Na quarta coluna estão os códigos dos fornecedores de cada componente com um link para o website onde será feita a compra e onde é possível obter informações detalhadas de cada componente. Na quinta coluna há uma breve descrição de cada componente.

Tabela 1 – Lista de Materiais

quantidade	posição	part number	codigo do fornecedor	descrição
1	U3	LMV321ILT	497-4941-1-ND	ampop
1	D1	DB2J40700L	DB2J40700LCT-ND	diodo schottky
1	3, 4, 5, 26, 27, 20, 21,	APT2012CGCK	754-1127-2-ND	led verde smd 805 2000 unidades
1	D15, D28, D29, D25,	1N4148W-TP	1N4148WTPMSTR-ND	diodo smd 1N4148 3000 unidades
1		PEC11L-4220F-S0015	PEC11L-4220F-S0015-ND	encoder incremento/decremento
2	Q3, Q4,	BC817-25LT1G	BC817-25LT1GOSCT-ND	transistor NPN comando
3	Q1, Q5, Q6,	MMBT4403	MMBT4403FSCT-ND	transistor PNP comando
1	j16	DM3D-SF	HR1941CT-ND	soquete micro sd
1	LS3	668-1454-ND	AI-1027-TWT-5V-2-R	buzzer
2	P1, P2	S9201-ND	SFH11-PBPC-D25-ST-BK	conector base fema arm
1	2, 73, 66, 13, 58, 59, 6	RMC0805JT10K0TR-ND	RMC0805JT10K0	resistor 10K 0805 5000 unidades

2.1.3: Teste de Blocos do Circuito

Antes do início do processo de desenvolvimento da PCI foram realizados alguns testes para validação de dois dos blocos representados no diagrama da Figura 5. O motivo de os testes não terem sido feitos nos demais blocos é que a maioria deles já havia sido validada pelo projetista do circuito, em projetos anteriores, e seu bom funcionamento já havia sido comprovado. Outros deles eram muito simples, como por exemplo ligações elétricas para pinos de conectores, e portanto não foram testados durante o desenvolvimento deste projeto.

Os dois blocos testados são responsáveis por manipulações mais críticas de sinais: condicionamento de sinais para medição e transmissão de sinais de comunicação serial.

2.1.3.1: Teste do Bloco Condicionamento de Sinais para Medição

O propósito do condicionamento do sinal para medição é que este seja transmitido para o conversor A/D localizado na Placa de Processamento, para então ser convertido e ter seu valor medido. O periférico responsável pela conversão possui características elétricas particulares e, se o valor de tensão do sinal elétrico de entrada desse periférico não tiver uma faixa de variação restrita a um intervalo particular, esse periférico pode ser danificado. O Bloco de Condicionamento faz com que o valor do sinal a ser convertido permaneça no intervalo seguro. Então o teste

desse bloco consistiu em enviar um sinal de medição em sua entrada, e medir o sinal resultante na saída.

O que pôde ser observado é que mesmo que o sinal na entrada tivesse um valor proibido em um determinado momento, nesse mesmo momento o circuito de condicionamento atuava no sentido de enviar para a saída o valor máximo permitido pelo periférico de conversão A/D, e o funcionamento desse bloco foi então validado.

2.1.3.1: Teste do Bloco Circuito para Comunicação Serial

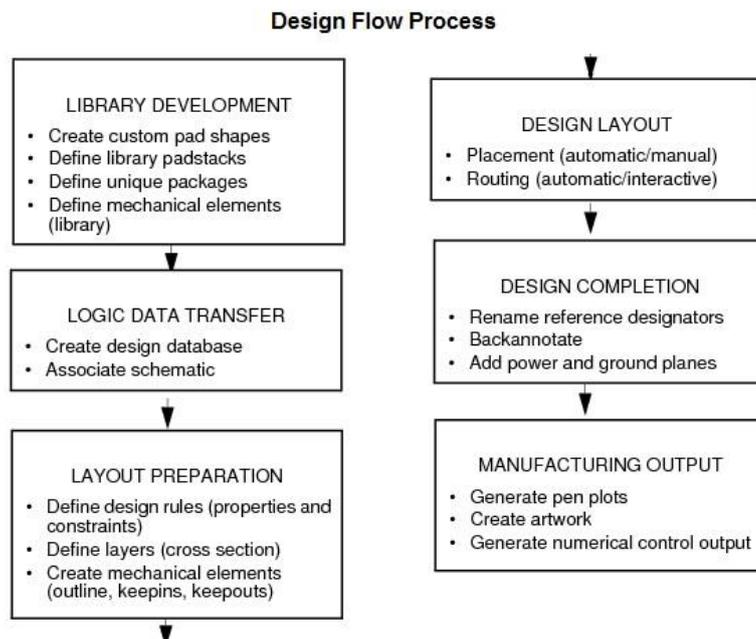
Os limites desejados do valor de frequência do sinal de comunicação serial para o sistema já estavam definidos. O teste desse bloco consistiu então em variar a frequência do sinal de comunicação na entrada do bloco e observar o sinal na saída.

O que pôde ser observado é que o bloco conservou a integridade do sinal mesmo a uma frequência superior a 5 vezes o limite desejado para a frequência do sinal, e isso foi parâmetro suficiente para a sua validação.

2.2: Fluxo de Desenvolvimento da PCI

O processo de desenvolvimento da PCI foi definido com base nas sugestões do Manual do Usuário do software CAD de desenvolvimento utilizado. A Figura 8 ilustra essas sugestões. [1]

Figura 8 – Fluxo de desenvolvimento da PCI.



Fonte: Allegro® UserGuide: GettingStartedwithPhysical Design. ProductVersion 16.6 October 2012

A seguir serão descritas com mais detalhes as etapas utilizadas no desenvolvimento da PCI.

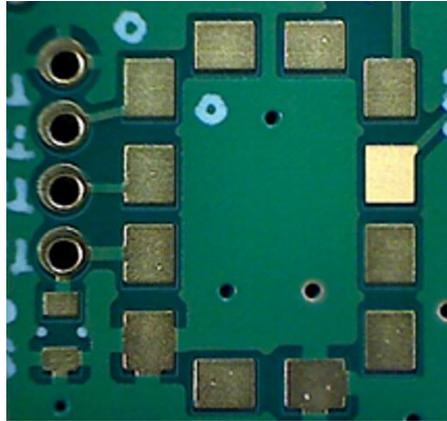
2.2.1: Desenvolvimento de Bibliotecas

O desenvolvimento de bibliotecas consiste em criar uma representação física de cada componente eletrônico cujo símbolo é representado no esquemático do circuito. Adicionalmente foi necessário utilizar representações para elementos mecânicos que não estão descritos no esquemático, mas que deverão estar presentes na PCI, como por exemplo elementos mecânicos de fixação da PCI.

2.2.1.1: Pads

Pads são regiões da PCI que serão posteriormente soldadas aos pinos dos componentes eletrônicos. Há dois tipos básicos de pads: os SMD que serão soldados a pinos que permanecem na superfície da PCI e os PTH que serão soldados a pinos que atravessam a PCI (Figura 9). Esses pads variam em forma e tamanho, entretanto neste projeto foram utilizados apenas o formato retangular, não sendo necessário projetar pads com formatos diferentes.

Figura 9 – Pads SMD e PTH.



Fonte: www.eurocircuits.com

Na ferramenta CAD utilizada os pads são projetados utilizando uma ferramenta denominada Pad Designer (Figuras 10 e 11).

A Figura 10 mostra a aba de parâmetros da ferramenta, onde são definidos parâmetros como unidades de medida, tamanho e símbolo de furos (pads PTH).

A Figura 11 mostra a aba de camadas. Essa aba varia conforme a opção de projeto de pad SMD e PTH. Para pads SMD é possível definir a geometria do pad (circular, retangular), o alívio térmico do pad e o anti-pad, bem como as camadas utilizadas (SOLDERMASK_TOP, PASTEMASK_TOP, FILMMASK_TOP).

O alívio térmico pode ser observado na Figura 9 no pad circular do topo superior esquerdo da figura, e consiste em não conectar todas as extremidades do pad. Isso permite um processo mais fácil de soldagem do componente elétrico, com uma temperatura menor, uma vez que diminui a transferência térmica entre o pad e o ambiente que o circunda.

O anti-pad é usado apenas quando a PCB possui camadas internas de cobre. Sua função é impedir que um pino de um componente PTH seja conectado às camadas internas de cobre. Essencialmente o anti-pad é a representação de um furo na camada interna de cobre que não deve ser conectada ao pino do

componente. Na PCI projetada não serão usados anti-pads uma vez que não haverá camadas internas de cobre.

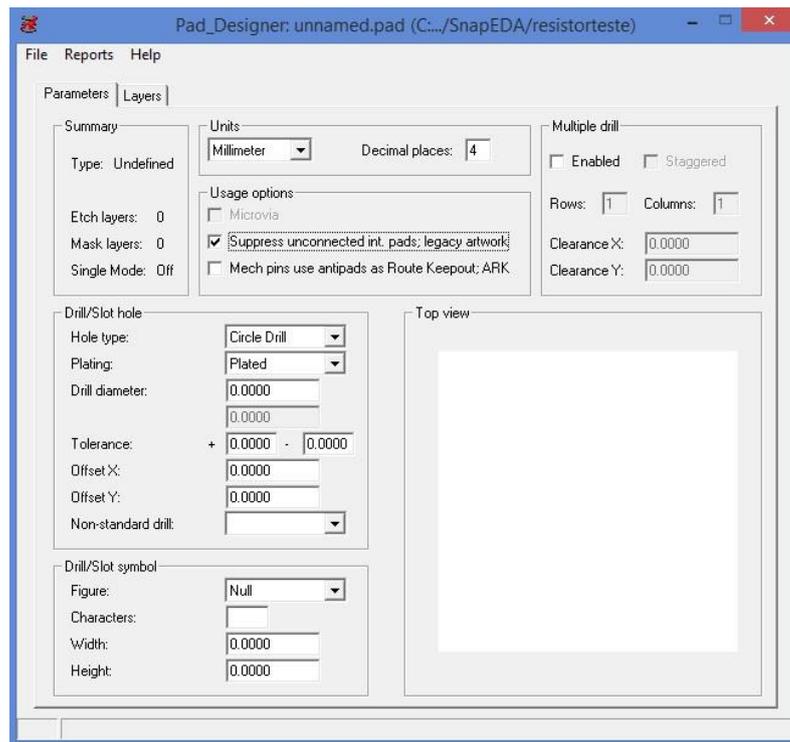
A camada `SOLDERMASK_TOP` permite definir o tamanho da sombra que será projetada sobre o pad durante a fabricação da PCI. Um dos processos finais da fabricação da PCI consiste em cobrir a placa com um material de proteção, semelhante a uma tinta. Esse material é fixado (colado) à placa através da exposição à luz, de modo que, havendo sombras na placa durante esse processo, o material isolante não será fixado. É necessário que haja sombra sobre as regiões da placa que deverão ser soldadas aos componentes, para que não haja o isolamento dessas regiões. As sombras utilizadas no projeto dos pads foram escolhidas com a mesma geometria e dimensões dos respectivos pads.

A camada `PASTEMASK_TOP` permite definir o tamanho do furo por onde passará a pasta de solda durante o processo de soldagem dos componentes. A técnica de soldagem dos componentes da PCI não usará pasta de solda, portanto não foi necessário a definição desse parâmetro.

A camada `FILMMASK_TOP` é uma camada genérica que não foi utilizada no projeto da PCI.

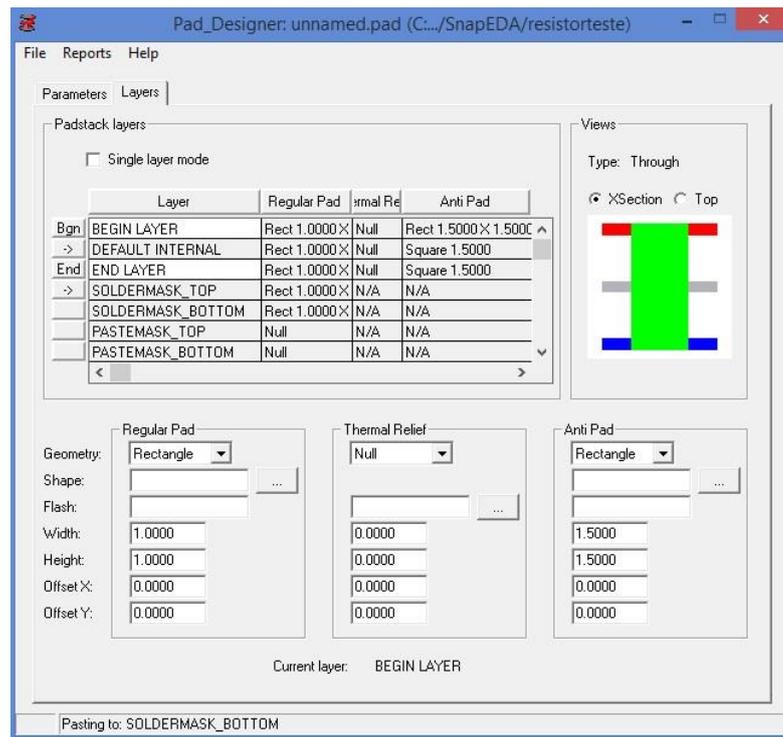
Na Figura 11 está mostrado o desenho da seção transversal de um pad PTH, com as camadas superior, inferior e interna, bem como o furo central.

Figura 10 – Pad Designer (aba de parâmetros).



Fonte: Allegro® Pad Designer, Version 16.6

Figura 11 – Pad Designer (aba de camadas).

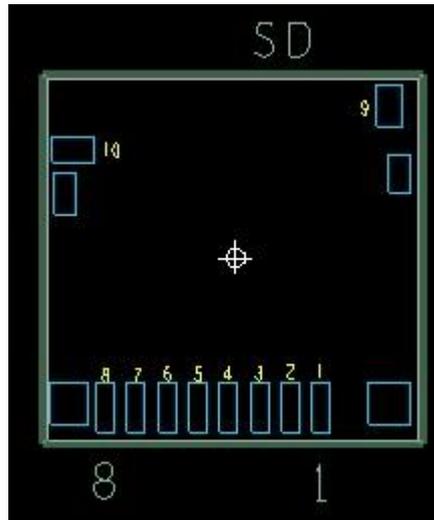


Fonte: Allegro® Pad Designer, Version 16.6

2.2.1.2: Definição de Packages

Packages são as representações físicas dos componentes. Por exemplo, para um conector para cartão SD essas representações podem ser vistas na Figura 12. Os packages definem as fronteiras de cada componente, a localização dos pads relacionados a cada componente, bem como a numeração desses pads e a referência do componente. Para a definição dos Packages, inicialmente é desenhada a fronteira do componente, em seguida são adicionados os pads previamente projetados e então é feito o posicionamento desses pads seguido da numeração apropriada. Define-se então a referência (SD na Figura 12).

Figura 12 – Package de um cartão SD.



Todas as informações necessárias para a definição do package de cada componente são obtidas do datasheet do componente que é indicado na lista de materiais. No caso do conector da Figura 12, as informações podem ser conseguidas no desenho técnico do componente localizado no seu datasheet. A Figura 13 mostra o desenho técnico do cartão SD utilizado.

Figura 13 – Desenho técnico do cartão SD utilizado.

Essas informações são obtidas do arquivo salvo ao final da criação de cada Package. Posteriormente será criado um “Netlist”, que é um arquivo de transferência que posteriormente será carregado no software CAD.

2.2.3: Preparação do Layout

Antes do projeto efetivo da PCI (posicionamento dos componentes e conexão das trilhas elétricas), é necessário passar por uma etapa de preparação, onde serão definidas informações importantes do projeto.

2.2.3.1: Definição de Regras de Design (Propriedades e Restrições)

Primeiramente são definidas informações de tamanho de página, tamanho de texto, unidades de medida utilizadas, tamanho do grid, etc. Em seguida deve-se fornecer informações sobre as restrições do projeto, como, por exemplo, larguras de trilhas e espaçamentos.

As informações referentes às restrições do projeto são obtidas da empresa que fabricará a PCI. A seguir serão detalhadas essas informações.

Informações Técnicas

Placas Metalizadas (Face Dupla)

1oz (1 oz / ft² = 0.035mm)

Largura mínima de trilhas: 8 mils

Espaçamento mínimo entre trilhas: 8 mils

Espaçamento mínimo entre trilhas e vias: 8 mils

Espaçamento mínimo entre trilhas e pads: 8 mils

Espaçamento mínimo entre pads: 8 mils

Espaçamento mínimo entre vias: 8 mils

Espaçamento mínimo entre trilhas e malha: 8 mils

Espaçamento mínimo entre pads e malha: 8 mils

Espaçamento mínimo entre vias e malha: 8 mils

Espaçamento mínimo de trilhas/pads e malhas para a borda: 10mils

Tamanho mínimo de furação: 0,3mm

8mils – 0,20mm

10mils – 0,25mm

Corrente elétrica nas trilhas: 200mA / mm de largura (valor típico)

Corrente elétrica nas vias: 250mA / via (supondo via 0.5mm)

Furo de vias = 10mils (0,25mm)

Obs.: para placas de 2oz considerar especificações de 12mils (0,30mm).

Obs.: para placas de 3oz considerar especificações de 16mils (0,40mm).

As informações detalhadas acima são particulares da empresa escolhida para a fabricação da PCI e variam de empresa para empresa.

A indicação de 1oz na primeira linha refere-se à espessura das camadas (superior e inferior) de material condutor (cobre). A PCI foi projetada utilizando a espessura 2oz, sendo necessário aplicar o ajuste informado na penúltima linha. A Figura 14 mostra o posicionamento das camadas de cobre da PCI.

Figura 14 – Camadas condutoras e isolante da PCI.



Os “fios” que conduzirão o sinal elétrico entre os componentes da PCI são denominados “trilhas”. Sendo necessário conectar uma trilha da camada condutora superior a uma outra trilha da camada condutora inferior, essa conexão será feita por meio de um furo na PCI. Esse furo é denominado “via” e possui paredes metalizadas para permitir a condução de corrente elétrica.

“Malha” é uma região da camada condutora semelhante a uma trilha (permite a conexão elétrica entre componentes), entretanto com área de superfície maior, podendo em alguns casos se estender por quase toda a superfície da PCI.

Nas **Informações Técnicas** acima, a linha “Tamanho mínimo de furação” refere-se a furos para pads PTH.

Uma vez obtidas da empresa que fabricará a PCI as informações referentes às restrições do projeto, deve-se transferir essas informações para as Regras de Design. Isso foi feito através da ferramenta Constraint Manager que é uma ferramenta interna do software CAD. Essa ferramenta consiste basicamente em um conjunto de tabelas onde podem ser inseridas as restrições obtidas da fabricante da PCI. Nas Figuras 15 e 16 a seguir estão ilustradas tabelas de espaçamentos e de espessuras.

Figura 15 – Exemplo de tabela de espaçamentos.

Objects		Referenced Spacing CSet	Line To									
Type	Name		Line	Thru Pin	SMD Pin	Test Pin	Thru Via	BB Via	Test Via	Shape	Bond Finger	Hole
			mm	mm	mm	mm	mm	mm	mm	mm	mm	mm
Dsn	V1-1-W19	DEFAULT	0.1270	0.1270	0.1270	0.1270	0.1270	0.1270	0.1270	0.1270	0.1270	0.2032
Net	A IN	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	A OUT	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	AD_ARM	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	B IN	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	B OUT	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	BE_IN	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	BUZZER	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	B_DIR	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	B_E	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	CD	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	CLK	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	CMD	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	CONTRASTE	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	DATA 0	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	DATA 1	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	DATA 2	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	DATA 3	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	DIR	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	DIR IN	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	DISP FONTE	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	DISP IN	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	DISPARO	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	DISP1	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	DISP2	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	D4	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	D5	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	D6	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	D7	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	E	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
Net	EN POT	DEFAULT	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000

Fonte: Allegro® Constraint Manager, Version 16.6

Figura 16 – Exemplo de tabela de espessuras.

Objects		Referenced Physical CSet	Line Width		Neck		
Type	S		Min	Max	Min Width	Max Length	
Name			mm	mm	mm	mm	
*	*	*	*	*	*	*	
Dsn	☐	V1-1-W19	DEFAULT	0.1270	0.0000	0.1270	0.0000
Net		A IN	DEFAULT	0.5000	0.0000	0.3000	50.0000
Net		A OUT	DEFAULT	0.5000	0.0000	0.3000	50.0000
Net		AD_ARM	DEFAULT	0.5000	0.0000	0.3000	50.0000
Net		B IN	DEFAULT	0.5000	0.0000	0.3000	50.0000
Net		B OUT	DEFAULT	0.5000	0.0000	0.3000	50.0000
Net		BE_IN	DEFAULT	0.5000	0.0000	0.3000	50.0000
Net		BUZZER	DEFAULT	0.5000	0.0000	0.3000	50.0000
Net	▶	B_DIR	DEFAULT	0.5000	0.0000	0.3000	50.0000
Net		B_E	DEFAULT	0.5000	0.0000	0.3000	50.0000
Net		CD	DEFAULT	0.5000	0.0000	0.3000	50.0000
Net		CLK	DEFAULT	0.5000	0.0000	0.3000	50.0000
Net		CMD	DEFAULT	0.5000	0.0000	0.3000	50.0000
Net		CONTRASTE	DEFAULT	0.5000	0.0000	0.3000	50.0000
Net		DATA 0	DEFAULT	0.5000	0.0000	0.3000	50.0000
Net		DATA 1	DEFAULT	0.5000	0.0000	0.3000	50.0000
Net		DATA 2	DEFAULT	0.5000	0.0000	0.3000	50.0000
Net		DATA 3	DEFAULT	0.5000	0.0000	0.3000	50.0000
Net	▶	DIR	DEFAULT	0.5000	0.0000	0.3000	50.0000
Net		DIR IN	DEFAULT	0.5000	0.0000	0.3000	50.0000
Net		DISP FONTE	DEFAULT	0.5000	0.0000	0.3000	50.0000
Net		DISP IN	DEFAULT	0.5000	0.0000	0.3000	50.0000
Net		DISPARO	DEFAULT	0.5000	0.0000	0.3000	50.0000
Net		DISP1	DEFAULT	3.0000	0.0000	2.4000	50.0000
Net		DISP2	DEFAULT	3.0000	0.0000	2.4000	50.0000
Net		D4	DEFAULT	0.5000	0.0000	0.3000	50.0000
Net		D5	DEFAULT	0.5000	0.0000	0.3000	50.0000
Net		D6	DEFAULT	0.5000	0.0000	0.3000	50.0000
Net		D7	DEFAULT	0.5000	0.0000	0.3000	50.0000
Net		E	DEFAULT	0.5000	0.0000	0.3000	50.0000
Net	▶	EN POT	DEFAULT	0.5000	0.0000	0.3000	50.0000

Fonte: Allegro® Constraint Manager, Version 16.6

Dentre as definições de regras de design, uma das mais importantes é a definição da espessura das trilhas. Há uma série de situações e detalhes que devem ser considerados. Por exemplo, as trilhas devem ser dimensionadas com espessura suficiente para conduzir a corrente elétrica que deverá passar por ela. Entretanto a corrente elétrica em cada trilha possui valores que variam durante o funcionamento do equipamento. Faz-se necessário então verificar o valor máximo de corrente elétrica que passará por uma trilha e definir a espessura dessa trilha com base no valor da corrente, incluindo possivelmente uma margem de segurança.

Apesar de a relação corrente/espessura para as trilhas ser um fator fundamental, esse fator não é o único. Há que levar em consideração os comprimentos das trilhas na PCI. Quanto maior o comprimento de uma trilha, maior

o valor da resistência elétrica dessa trilha, e isso pode influenciar consideravelmente fatores como queda de tensão elétrica e dissipação de potência. Se a trilha não for dimensionada para suportar a corrente elétrica que passará por ela, poderá ocorrer o rompimento da trilha, comprometendo a PCI.

Uma maneira de dimensionar uma trilha que conecta o Componente 1 (C1) com o Componente 2 (C2) é verificar primeiramente qual é o valor de corrente elétrica máxima que cada um dos componentes suporta. Por exemplo, considerando que C1 suporta uma corrente máxima de 1A e C2 uma corrente máxima de 2A, então, se a trilha for dimensionada para suportar uma corrente elétrica máxima de 1A e essa corrente surgir, haverá o risco de o componente 1A ser danificado e também de a trilha ser rompida. Já se a trilha for dimensionada para suportar uma corrente elétrica máxima de 2A e a corrente elétrica superar o valor de 1A, então o componente 1A será danificado, interrompendo a transmissão da corrente e protegendo a integridade da trilha e do componente 2A. É importante dar preferência à integridade da trilha da PCI, pois se ela for rompida então toda a PCI pode ser comprometida, ao ponto que se ocorrer a danificação de apenas um componente eletrônico da PCI, esse componente pode ser trocado e a PCI como um todo tem mais chances de ser protegida.

Do ponto de vista unicamente de proteção da integridade da trilha e dos componentes conectados a ela, o exemplo acima, onde a trilha foi dimensionada para suportar uma corrente elétrica de 2A, não faz tanto sentido, uma vez que a corrente nessa trilha nunca chegará a 2A (o componente 1A será danificado e interromperá a passagem da corrente). Essa trilha poderia ter sido dimensionada para suportar correntes de valores menores que 2A. Entretanto há outros pontos de vista a considerar, como por exemplo a capacidade/incapacidade de a fabricante fabricar trilhas de espessuras menores (discutido anteriormente) e a densidade da PCI.

A densidade da PCI é outro fator a ser considerado. Há projetos em que a PCI deverá ter o tamanho reduzido ao máximo para poder ser posicionada em ambientes de espaço limitado, devendo-se então projetar trilhas com espessuras de valores mínimos possíveis. Há projetos em que isso não é um fator crítico e, para

esses, pode-se até mesmo projetar trilhas cujos valores de espessura são bem maiores do que o necessário para os componentes que são conectados por elas.

Neste projeto foi utilizada uma regra simples: trilhas de sinal com espessura de valor 0.5mm e trilhas de potência com espessura de valor 3mm. Trilhas de sinal são trilhas que serão percorridas por correntes elétricas de valor em torno de 20mA, e trilhas de potência são trilhas que serão percorridas por correntes elétricas de valor superior a 1A.

2.2.4: Desenvolvimento do Layout

A etapa de desenvolvimento do layout da PCI é dividida em duas partes: Posicionamento dos componentes e roteamento das trilhas.

Essencialmente há duas maneiras de realizar essas duas partes. A primeira maneira consiste em realizar o posicionamento de todos os componentes e, após isso, iniciar e finalizar o roteamento. A segunda maneira consiste em realizar o posicionamento de parte dos componentes, fazer o roteamento adequado dessa parte, para então posicionar uma segunda parte e fazer o roteamento dessa segunda parte, e assim sucessivamente. Seja qual for a maneira escolhida, o software CAD utilizado permite a utilização de uma ferramenta para posicionamento automático dos componentes, bem como uma outra ferramenta para o roteamento automático das trilhas.

A estratégia de posicionamento e roteamento utilizada neste projeto foi a seguinte: posicionar uma parte dos componentes e realizar o roteamento das trilhas correspondentes a essa parte, em seguida posicionar uma segunda parte de componentes e realizar o roteamento dessa segunda parte, e assim sucessivamente. Além disso, tanto o posicionamento dos componentes como o roteamento das trilhas foram feitos manualmente.

A seguir serão fornecidos mais detalhes correspondentes ao posicionamento dos componentes e ao roteamento das trilhas elétricas.

2.2.4.1: Posicionamento dos Componentes e Roteamento das Trilhas Elétricas

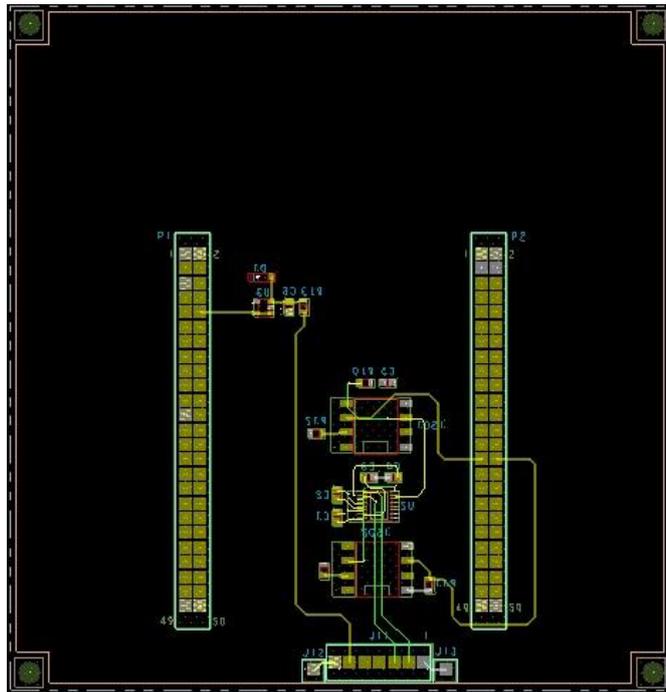
Primeiramente foram desenhados os limites da PCI, utilizando uma geometria retangular. A seguir foram posicionados os elementos de fixação mecânica (furos para parafusos) nos vértices da PCI. Depois foi desenhada uma nova geometria, denominada “RouteKeepin”, interna à geometria retangular da PCI, e essa nova geometria exclui os elementos de fixação dos vértices. A função da região “RouteKeepin” é impedir que qualquer trilha seja roteada para o exterior dessa região.

Após a definição dessa geometria “RouteKeepin”, inicia-se o posicionamento e o roteamento do primeiro conjunto de componentes, como mostrado na Figura 17.

Uma observação importante a respeito das cores nas imagens a seguir: alguns componentes possuem pads com cor amarela, e isso indica que esses componentes estão posicionados na face inferior da PCI. Outros possuem pads com cor verde, e isso indica que esses componentes estão posicionados na face superior da PCI. Há, no entanto, uma exceção: há componentes cujos pads têm cores que variam de uma imagem para outra, e isso indica que esses componentes possuem pads do tipo PTH. Componentes cujos pads estão ilustrados sempre com a mesma cor são componentes SMD.

Há uma série de detalhes a serem considerados durante o posicionamento dos componentes: facilidade de acesso aos conectores, tamanho dos componentes, nível dos sinais elétricos que percorrem os componentes, acoplamento futuro com outras PCIs, acessibilidade para medição de sinais elétricos, facilidade de visualização de LEDs indicadores, facilidade na soldagem dos componentes, entre outros. Para tratar esses detalhes é importante ter uma visão geral do equipamento como um todo do ponto de vista do seu funcionamento, e não apenas da PCI. Para isso é fundamental obter a orientação dos profissionais envolvidos no desenvolvimento do projeto mecânico, no desenvolvimento de outros blocos de hardware e até mesmo dos profissionais envolvidos no desenvolvimento do software.

Figura 17 – Posicionamento e roteamento do primeiro conjunto de componentes.



A seguir o processo de posicionamento e roteamento é repetido com um novo conjunto de componentes, como mostrado na Figura 18. Nas Figuras 19 e 20 isso é feito novamente. Na Figura 21 o posicionamento dos componentes está finalizado, entretanto o roteamento ainda não está completo, uma vez que há pads que só serão conectados através de malhas em uma etapa posterior de finalização do layout.

Figura 18 – Posicionamento e roteamento do segundo conjunto de componentes.



Figura 19 – Posicionamento e roteamento do terceiro e quarto conjuntos de componentes.

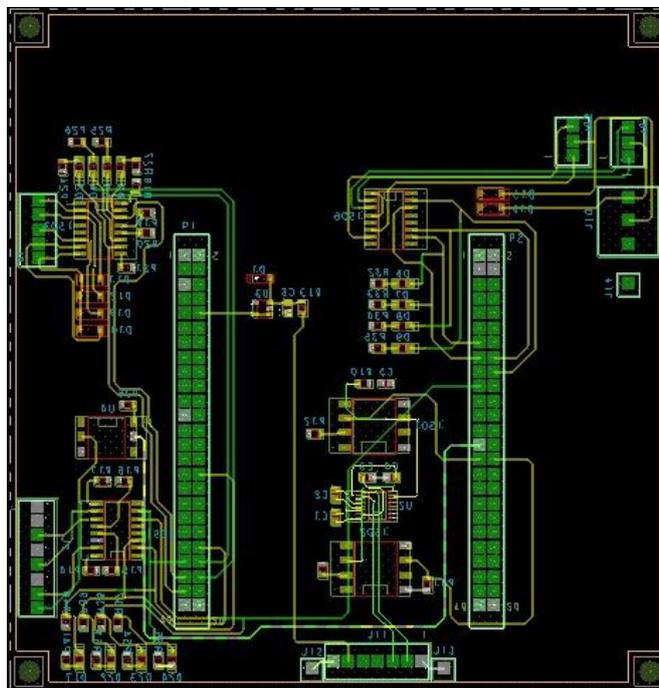


Figura 20 – Posicionamento e roteamento do quinto e sexto conjuntos de componentes.

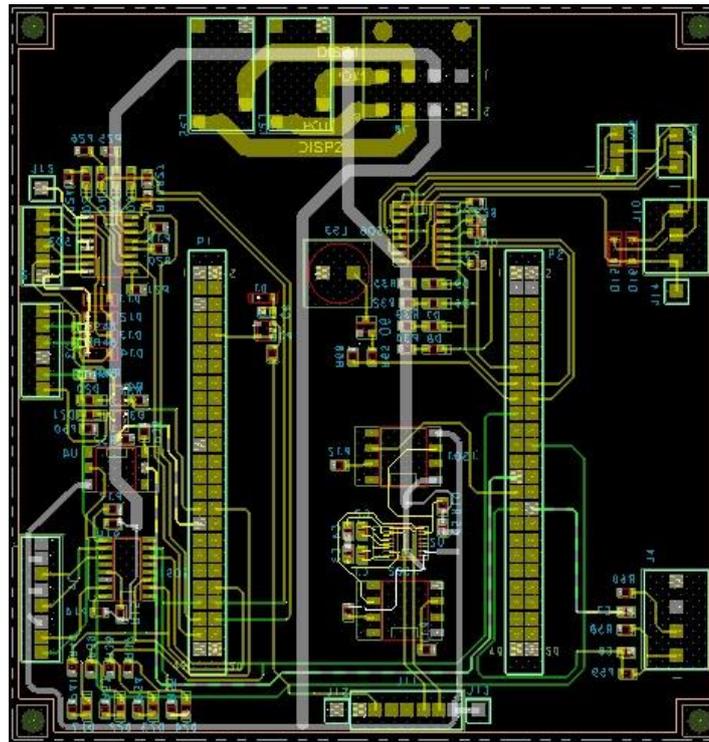
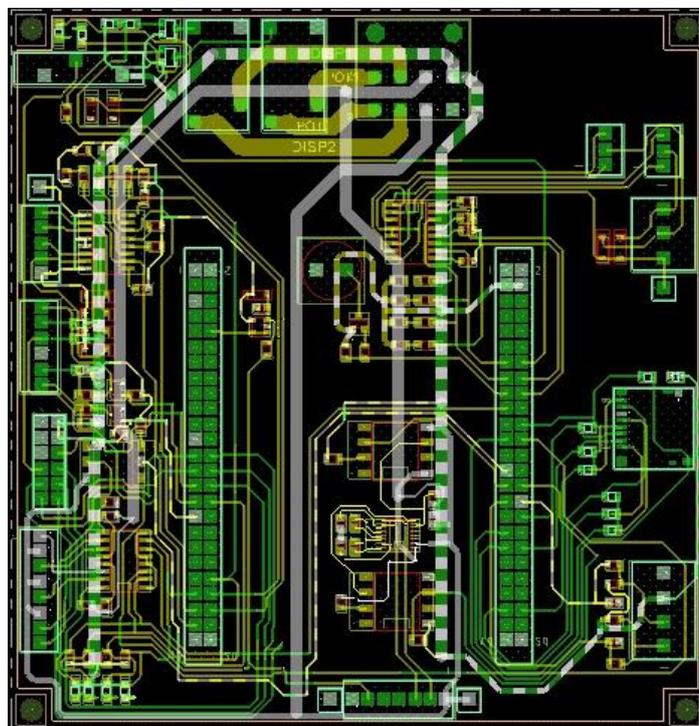


Figura 21 - Posicionamento e roteamento do sétimo conjunto de componentes.



2.2.5: Finalização do Layout

Na etapa de finalização do layout existe a possibilidade de alterar os nomes (referências) dos componentes e em seguida sincronizar essa alteração com o esquemático do circuito elétrico. Para este projeto isso não foi necessário.

Na Figura 22 estão mostrados os componentes adequadamente posicionados e suas referências. As trilhas elétricas foram ocultas para melhor visualização. Os pequenos pontos isolados representam as vias.

Na etapa de finalização do layout também são incluídas as malhas (planos de GND e planos de potência). Neste projeto foram incluídas apenas malhas de GND (superior e inferior). Na Figura 23 está ilustrada a malha de GND superior e na Figura 24 a malha de GND inferior.

Figura 22 – Visão do posicionamento completo dos componentes (trilhas ocultas para facilitar a observação).



Figura 23 – Malha de GND superior.

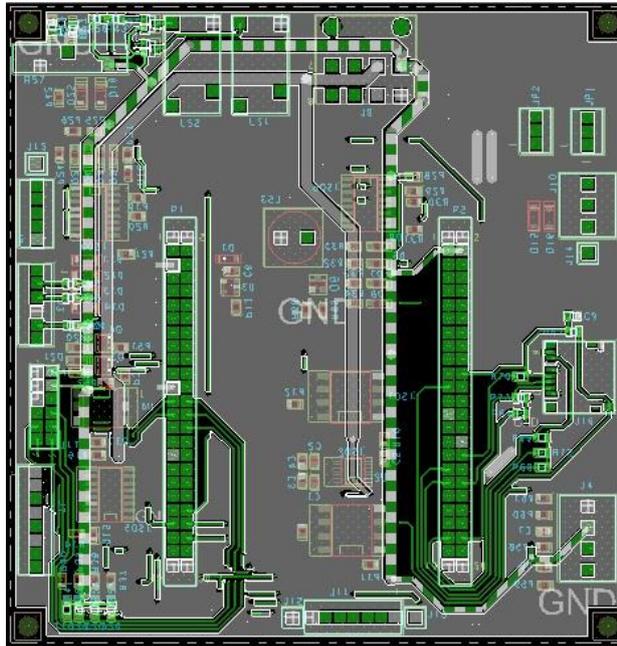
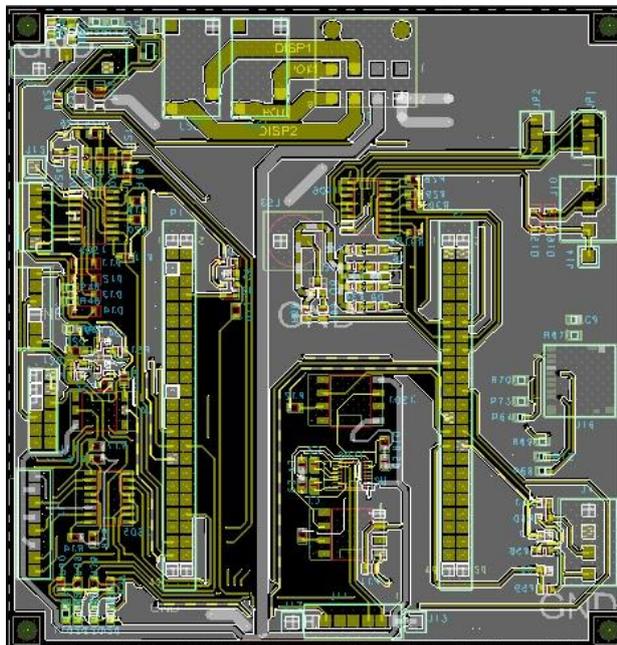


Figura 24 – Malha de GND inferior.



2.2.6: Arquivos de Fabricação

Nesta etapa o software CAD utilizado, mediante configurações apropriadas, é capaz de gerar os arquivos necessários para que a PCI possa ser fabricada.

Dentre esses arquivos estão os arquivos para o desenho dos pads e das trilhas das camadas superior e inferior, arquivos com a localização dos furos que deverão ser feitos na PCI e arquivos contendo as localizações das referências dos componentes.

Empresas de fabricação de PCI diferentes podem usar processos diferentes de fabricação, o que pode requerer tipos diferentes de arquivos. Deve-se então obter essas informações da empresa escolhida e configurar o software CAD para gerar tipos apropriados de arquivos

Capítulo 3: Discussões Sobre Organização do Software

A PCI cujo desenvolvimento foi descrito no Capítulo 3 precisará ser conectada a uma Placa de Comando e Controle para que o equipamento de soldagem possa funcionar. Essa Placa de Comando e Controle possui certamente um elemento computacional (microcontrolador) para o qual deverá ser desenvolvido um software.

Existem diferentes maneiras e abordagens de organização de um software, dentre elas estão a abordagem utilizando interrupções e a abordagem concorrente, que serão brevemente discutidas a seguir.

As discussões a seguir são genéricas e não estão acopladas a um hardware específico. Assim, não será necessário entrar em detalhes de configurações de periféricos. Esses detalhes estão já documentados nos manuais dos hardwares utilizados.

3.1: Abordagem Utilizando Interrupções

Nesse caso o software consiste basicamente em uma etapa de configuração, onde as variáveis são inicializadas e os periféricos do microcontrolador são configurados, e uma etapa de execução, onde as instruções são executadas e a lógica de operação é implementada.

Essa segunda etapa pode ser implementada em um loop infinito.

O sistema desenvolvido é reativo, ou seja, ele executa instruções com base em informações que observa e recebe do ambiente. Por exemplo, o sistema pode receber informações a qualquer momento de um operador externo por meio de uma IHM. É necessário então implementar mecanismos que permitam ao microcontrolador monitorar canais de entrada de informações. Um desses mecanismos pode ser, por exemplo, uma linha de instrução onde é feita a leitura do sinal binário de entrada em um pino. O sistema também pode enviar informações

para o ambiente externo por meio de uma tela de LCD. É necessário então implementar mecanismos que permitam ao microcontrolador enviar informações para o LCD. Um desses mecanismos pode ser, por exemplo, uma sequência de instruções que alterem os estados lógicos binários de saída de um conjunto de pinos do microcontrolador conectados ao circuito do LCD.

Para sistemas muito simples, onde existe pouca troca de informações entre o ambiente e o sistema, pode ser aceitável implementar esses mecanismos de troca de informações dentro da etapa de execução, no loop infinito. Entretanto para sistemas mais complexos essa forma de implementação não é adequada pois permite pouca eficiência no uso do processador. Essa baixa eficiência se reflete consideravelmente na tarefa de administração do tempo, pois cabe ao desenvolvedor do software toda a tarefa de analisar valores de frequências de clock, dividir esses valores por meio de incremento de registradores, criar contadores e verificar continuamente seus valores. Ou seja, fica a cargo do desenvolvedor tarefas que exigem um tempo e um esforço consideráveis, que poderiam ser investidos na tarefa principal do software que é fazer o equipamento realizar as tarefas do processo de soldagem.

Uma maneira de reduzir esse problema é implementando interrupções. Isso é feito configurando o microcontrolador para que em intervalos definidos de tempo, por exemplo, 10ms, 100ms ou até mesmo 1us, o fluxo de execução de instruções saia do loop infinito temporariamente para executar uma rotina lateral. Essa rotina lateral pode ser, por exemplo, o envio de informações a um LCD, a leitura do estado lógico binário de um pino ou a conversão de um sinal analógico para digital.

Outra questão a ser considerada é o gerenciamento das prioridades das interrupções utilizadas. Essa questão se torna evidente quando estudamos o seguinte caso: Inicia-se a execução de um ciclo de tratamento da interrupção 1, que tem a função de enviar informações para uma tela de LCD. Antes de esse ciclo terminar, ocorre uma interrupção 2. O que fará o processador? Interromperá o tratamento da interrupção 1 para iniciar o tratamento da interrupção 2 ou terminará o ciclo que já estava executando para, após isso, iniciar o tratamento do ciclo da interrupção 2? Como o processador deveria se comportar se a interrupção 2 estiver

relacionada a uma rotina de emergência? E se a interrupção 2 estiver relacionada à atualização de estados de um conjunto de LEDs?

Evidentemente é preciso desenvolver um mecanismo para atribuir adequadamente as prioridades para as interrupções, o que está fortemente relacionado ao funcionamento do equipamento e do processo de soldagem que está sendo executado.

Uma maneira de iniciar a organização do software utilizando interrupções pode ser listando as diferentes tarefas que devem ser executadas. Por exemplo, envio/recepção de informações por meio de comunicação serial, leitura de um sinal analógico e posterior conversão para digital, leitura de botões, atualização de LCD, atualização de LEDs, evolução da lógica de estados do sistema, leitura/escrita em um cartão SD, etc. Em seguida, atribuindo prioridades a essas tarefas e, posteriormente configurando uma interrupção para cada tarefa.

A organização do software utilizando interrupções representa um grande incremento na eficiência do uso do processador em comparação com usar unicamente um loop infinito, e é uma excelente solução para um conjunto maior de aplicações.

Uma possível limitação de uma abordagem utilizando interrupções é que pode haver um número reduzido de interrupções disponíveis para o microcontrolador utilizado, e essas interrupções estão geralmente associadas a periféricos (hardware) do microcontrolador, como por exemplo temporizadores, cuja quantidade também pode ser limitada. Se o microcontrolador possuir poucas interrupções, então pode existir a necessidade de tratar várias tarefas dentro de uma mesma interrupção, o que pode prejudicar o desempenho do sistema.

3.2: Abordagem Concorrente Utilizando um Sistema Operacional

Nesse caso o software consiste em uma etapa de configuração, onde o sistema operacional é configurado, as variáveis são inicializadas, os periféricos do

microcontrolador são configurados e as tarefas (tasks) são definidas. A etapa seguinte seria implementar a lógica de cada tarefa.

A definição de uma tarefa contém informações sobre sua prioridade e sua periodicidade. A periodicidade indica a frequência de execução da tarefa. As prioridades e periodicidades das tarefas são questões fundamentais e estão diretamente ligadas à lógica de funcionamento do equipamento.

Uma vantagem interessante da abordagem concorrente utilizando um sistema operacional é que o desenvolvedor necessita apenas informar a periodicidade (por exemplo, 100 Hz) e informar a prioridade (por exemplo, prioridade 0, ou 5). Nessas questões a tarefa do desenvolvedor é informar (definir), e não implementar. O sistema operacional é que se encarrega da implementação. E isso permite uma economia de tempo e um maior esforço focado em questões mais operacionais do sistema, como a lógica de funcionamento.

Uma vez concluída a etapa de configuração e de definição e implementação das tasks, o sistema pode ser executado, e essa execução consiste na chamada do escalonador do sistema operacional.

O escalonador é um componente do sistema operacional responsável por gerenciar o uso do microprocessador pelas tarefas, respeitando as definições de prioridades e de periodicidades.

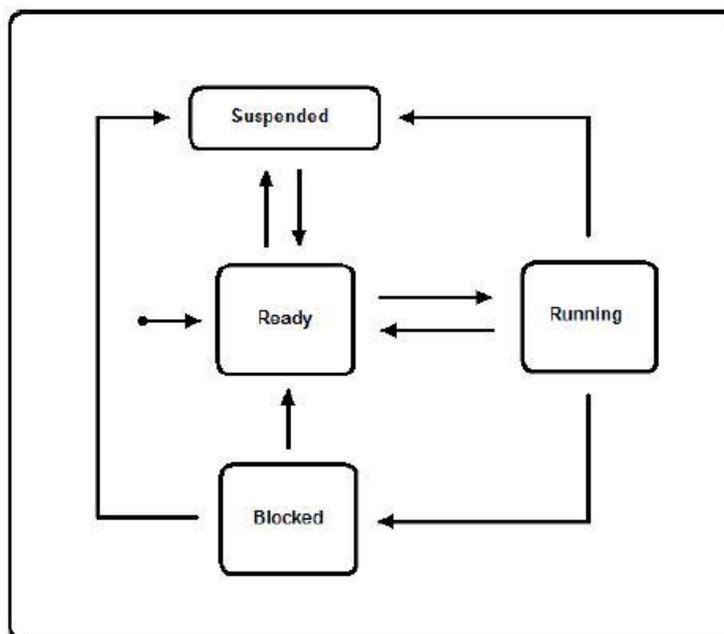
Em microprocessadores com mais de um núcleo é possível fazer com que duas tarefas sejam executadas simultaneamente. Em microprocessadores com apenas um núcleo, essa possibilidade não existe. O que é feito pelo sistema operacional no caso de um único núcleo é administrar o uso do microprocessador pelas tarefas, fazendo com que elas obtenham o microprocessador alternadamente.

Essa alternância no uso do microprocessador pelas tarefas faz com que seja natural a classificação delas em estados. A Figura 25 mostra estados possíveis das tarefas em um sistema operacional.

Supondo um sistema com 5 tarefas e um microprocessador com apenas um núcleo, apenas uma tarefa por vez poderá estar no estado “Running”, ou seja, o

estado no qual ela é executada pelo microprocessador. As outras 4 tarefas poderão estar em qualquer um dos outros 3 estados.

Figura 25 – Estados possíveis das tarefas em um sistema operacional.



As prioridades das tarefas influenciam na maneira com que o estado de cada uma dessas tarefas muda e também na maneira com que cada uma é implementada. Supondo novamente 5 tarefas e apenas uma tarefa podendo ser executada por vez (microprocessador de um único núcleo), e supondo também que essas tarefas possuem a mesma prioridade e que seus estados mudam apenas de “Running” para “Ready” ou de “Ready” para “Running”, o sistema operacional deverá então, com base em uma frequência determinada por um parâmetro de configuração, mudar os estados das tarefas de modo que todas elas possam obter o processador. Se a frequência com que ocorre a troca de estado for suficientemente grande considerando uma determinada aplicação, então no sistema como um todo haverá a ilusão de que todas essas tarefas estão sendo executadas ao mesmo tempo, quando na verdade elas estão sendo executadas uma por vez, concorrendo pelos recursos do hardware.

Supondo mais uma vez 5 tarefas, mas agora 4 delas possuindo a prioridade 1 e a última possuindo a prioridade 2, e supondo novamente apenas 2 estados

possíveis, o “Ready” e o “Running”, então uma vez que o processador iniciar a execução da tarefa de prioridade 2 essa tarefa não mais passará para o estado “Ready” e as outras tarefas não terão mais a possibilidade de serem executadas. Essa não é uma situação desejável.

O que é feito em sistemas com tarefas de prioridades diferentes é incluir em um determinado local do código de implementação das tarefas um comando para que elas passem para o estado “Blocked” e fiquem nesse estado por um período de tempo, ou até que um determinado evento aconteça, permitindo assim que tarefas de menor prioridade sejam executadas.

Outra vantagem do uso de um sistema operacional é que ele pode facilitar a execução de tarefas de tempo real, ou seja, tarefas que devem, obrigatoriamente, ser executadas com determinada frequência. Isso facilita o desenvolvimento de software para sistemas que precisam gerenciar o tempo de forma rigorosa.

Uma possível alternativa de sistema operacional é o FreeRTOS. Esse sistema operacional é gratuito, mesmo quando usado em aplicações comerciais [3].

Capítulo 4: Discussão dos Resultados

Durante o início do desenvolvimento da PCI, foi escolhido desenvolver bibliotecas de componentes manualmente. Embora alguns dos componentes definidos no esquemático do circuito possuam packages com geometria e pinagem muito comuns no mundo da eletrônica, e por isso já existirem bibliotecas prontas com esses componentes incluídos, mesmo assim optou-se por desenvolver essas bibliotecas manualmente para ter um entendimento melhor do processo de desenho. Pôde-se observar que é possível desenhar os packages obedecendo rigorosamente ao seu desenho no datasheet sem muito esforço, apesar de que para os primeiros componentes o tempo foi consideravelmente maior. Desenhar os packages manualmente, criando uma biblioteca particular para o projeto, permite mais flexibilidade porque ao final do processo o desenvolvedor tem pleno conhecimento dos elementos do desenho, bem como dos arquivos que o compõem e suas localizações.

A primeira etapa do desenvolvimento dos packages foi o desenho dos pads. Observou-se que o desenho dos pads não era tão simples como imaginado no início, pois há muitos detalhes a serem considerados, como por exemplo, camadas superior e inferior de pads PTH e camadas de máscaras de solda.

O desenvolvimento das bibliotecas ocupou cerca de 40% do tempo do projeto, sendo o tempo restante quase todo dividido entre as tarefas de posicionar os componentes adequadamente e fazer o roteamento das trilhas elétricas.

O esquemático obtido no início do desenvolvimento sofreu mudanças durante o processo de desenvolvimento da PCI, e essas mudanças tiveram que ser acomodadas no decorrer desse desenvolvimento. Percebeu-se que iniciar o projeto da PCI tendo posteriormente que acomodar atualizações do esquemático gera a necessidade de repetição de tarefas, desperdiçando tempo. O projeto como um todo fluiu de maneira consideravelmente mais fácil quando o esquemático não é alterado.

Notou-se que, para a etapa de posicionamento dos componentes na PCI, é fundamental estar em constante comunicação com a equipe de pessoas que desenvolve o projeto mecânico do equipamento e demais elementos de hardware. Isso porque há componentes que não devem ser acomodados em determinadas posições, pois, por exemplo, as dimensões desses componentes podem colidir com paredes internas da estrutura mecânica, ou podem não ficar facilmente visíveis, no caso de LEDs e conectores.

Há a possibilidade do posicionamento automático dos componentes, entretanto observou-se que, devido à grande quantidade de restrições de posicionamento dos componentes neste projeto, a configuração das ferramentas de posicionamento automático exigiria um tempo que não estava disponível.

O posicionamento dos componentes e o roteamento das trilhas foi feito em etapas pequenas, utilizando primeiro um pequeno conjunto de componentes, em seguida outro conjunto e assim sucessivamente. Essa utilização de pequenas etapas e pequenos blocos de circuito permitiu uma boa organização da PCI, mantendo componentes de subcircuitos semelhantes próximos uns dos outros. Essa proximidade entre componentes de mesmo subcircuito facilita o teste e possíveis manutenções futuras da PCI.

Da mesma forma que o posicionamento dos componentes, o roteamento das trilhas foi feito manualmente. Isso porque, também, a configuração das ferramentas de roteamento automático exigiria um tempo indisponível.

Com relação às discussões sobre software, observou-se que é possível, para o equipamento relacionado, desenvolver o software tanto utilizando interrupções quanto um sistema operacional. O sistema não é complexo o bastante para inviabilizar um projeto de software por interrupções. Entretanto a opção de utilizar um sistema operacional pode ser melhor do ponto de vista de futuras modificações, manutenções e expansões do sistema, uma vez que permite que o software fique menos complexo. Além disso, um sistema operacional pode ser capaz de gerenciar tempos críticos de uma maneira muito mais simples e mais segura do que um software projetado por interrupções.

Capítulo 5: Conclusões e Perspectivas

É natural que um projeto de hardware e software sofra atualizações com o passar do tempo. A placa de controle utilizada possui um microcontrolador poderoso, que é capaz de realizar muito mais tarefas do que o número de tarefas que a ele serão atribuídas neste sistema. Portanto pode ser viável, futuramente, adicionar mais subcircuitos à PCI, permitindo que ela realize mais tarefas e seja ainda mais flexível.

Bibliografia:

[1] “Alegro® User Guide”, Version 16.6, October 2012.

[2] www.labsolda.ufsc.br, Acesso em Agosto de 2016.

[3] www.freertos.org, Acesso em Agosto de 2016.

[4] www.st.com, Acesso em Agosto de 2016.