



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

- Grybot -

Um Robô Jogador Automático de Guitar Hero sobre FPGA

Luiz Adriano Andrade Oliveira Vaz

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientador
Prof. Dr. Marcus Vinicius Lamar

Brasília
2013

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Coordenador: Prof.^a Dr.^a Maristela Terto de Holanda

Banca examinadora composta por:

Prof. Dr. Marcus Vinicius Lamar (Orientador) — CIC/UnB
Prof. Dr. Ricardo Pezzoul Jacobi — CIC/UnB
Prof.^a Dr.^a Carla Denise Castanho — CIC/UnB

CIP — Catalogação Internacional na Publicação

Vaz, Luiz Adriano Andrade Oliveira.

- Grybot -: Um Robô Jogador Automático de Guitar Hero sobre FPGA
/ Luiz Adriano Andrade Oliveira Vaz. Brasília : UnB, 2013.

79 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2013.

1. jogos, 2. GuitarHero, 3. FPGA, 4. robô

CDU 004.4

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília-DF — Brasil

Dedicatória

Dedico aos meus pais que, mesmo morando distante, sempre acompanharam meus estudos e me ofereceram condições suficientes para concluir o curso de graduação.

Agradecimentos

Agradeço a Deus pela vitória de mais uma etapa da minha vida. Agradeço ao meu pai, Washington Luiz Bento de Azevedo, por seu exemplo e pelo conhecimento transmitido a mim diariamente, decisivo para a formação da minha personalidade e construção da minha vida profissional. Agradeço a minha mãe, Tânia da Silva Vaz, pelo carinho, cuidado e compreensão intermináveis e essenciais para manter-me emocionalmente equilibrado. Agradeço a minha irmã, Lara Andrade Oliveira Vaz pela sua preocupação e atenção comigo, deixando claro que nela posso sempre confiar. Agradeço a minha namorada, Priscila Oliveira Campos pelo imenso amor, carinho e apoio. Agradeço ao meu professor orientador, Marcus Vinicius Lamar pela confiança depositada em mim, pela liberdade de expor minhas idéias, pela sua paciência, por sua disponibilidade em realizar de reuniões, pelas cobranças e pela ótima orientação. Agradeço a todos meus amigos pela disposição de ouvir, aconselhar e me ajudar sempre que precisei.

Resumo

Jogos eletrônicos conquistaram espaço no meio acadêmico à medida que se tornaram parte de pesquisas na área de Inteligência Artificial e Robótica. *Guitar Hero*^{®3} (GH3) é um jogo de gênero musical cuja dificuldade representa um grande desafio para o jogador e exige muito treino para alcançar bons resultados. Sob a hipótese de que o processamento do sinal de vídeo sobre FPGA juntamente com um sistema eletromecânico de acionamento do controle do videogame é capaz de concluir todas as músicas do jogo no nível *Expert*; foi construído um robô jogador automático de GH3 chamado Grybot cujo desenvolvimento é detalhado neste trabalho. O protótipo construído atinge excelentes resultados da ordem de 97% de acertos nas músicas do jogo GH3 em todos os níveis de dificuldade.

Palavras-chave: jogos, GuitarHero, FPGA, robô

Abstract

Electronic games acquired more and more space in academia as they became part of researches in Artificial Intelligence and Robotics. Guitar Hero 3 (GH3) is a musical genre game in which difficulty represents a great challenge to the player and requires a lot of training to achieve good results. Under the assumption that video signal processing in FPGA together with a electromechanical system for triggering the videogame remote is capable of concluding all the songs in the game on Expert level; an automatic GH3 player robot called Grybot was constructed where the development is detailed in this thesis. The prototype constructed achieve excelent results on the order of 97% of success in the songs of the game in all difficulty levels.

Keywords: games, GuitarHero, FPGA, robot

Sumário

1	Introdução	1
1.1	Contextualização	1
1.2	Definição do Problema	2
1.3	Hipótese	2
1.4	Objetivos	3
1.5	Organização do Trabalho	3
2	Revisão Bibliográfica	4
2.1	Projetos de Automação de Jogos em Geral	5
2.1.1	Robô Jogador de Damas	5
2.2	Projetos de Automação do Jogo <i>Guitar Hero</i>	6
2.2.1	HeroNoid	6
2.2.2	SlashBot	8
2.2.3	CythBot	10
2.2.4	eGOR	11
2.2.5	DeepNote TM	12
2.2.6	AGH1000	14
2.2.7	Outros	15
3	Processamento de Sinais sobre FPGA	17
3.1	<i>Field Programmable Gate Array</i> (FPGA)	17
3.2	Vídeo Composto	20
3.3	<i>Video Graphics Array</i> (VGA)	21
3.4	Módulo DE2 TV	23
4	Modelo Robótico Proposto	25
4.1	Descrição do Problema	25
4.1.1	Captura do sinal de vídeo	25
4.1.2	Detecção de notas	26
4.1.3	Sincronização temporal	28
4.1.4	Calibração	28
4.1.5	Interação	29
4.2	Descrição do Projeto	29
4.2.1	Detecção de Cores	29
4.2.2	Temporização	32
4.2.3	Configuração e Monitoramento	37
4.2.4	Sistema de Acionamento	37

4.3	Grybot	39
5	Resultados	42
5.1	Análise do Sistema Mecânico	42
5.2	Definição dos Parâmetros	46
5.3	Desempenho Obtido	51
5.4	Comparações	58
5.4.1	Grybot vs Slashbot	58
5.4.2	Grybot vs Cythbot	59
5.4.3	Grybot vs Titan	59
5.4.4	Grybot vs eGOR	60
5.4.5	Grybot vs DeepNote	60
5.4.6	Grybot vs AGH1000	61
5.4.7	Grybot vs Seres Humanos	61
6	Conclusões	63
	Referências	65

Lista de Figuras

1.1	Tela do jogo <i>Guitar Hero</i> ^{®3}	2
2.1	Modelo representativo da máquina “O Turco” [37]	4
2.2	(a) Tabuleiro e câmera (b) Iluminação [21]	5
2.3	Peças peão vermelho e rei preto [21]	6
2.4	Robô HeroNoid [27]	6
2.5	(a) Região de captura das notas (b) Representação binária da imagem [27]	7
2.6	Robô SlashBot [7]	8
2.7	Sistema PXI [18]	9
2.8	(a). Solenoides (b). Circuito analógico de amplificação [7]	10
2.9	Robô CythBot [16]	10
2.10	Robô eGOR [22]	11
2.11	Regiões de detecção de notas [22]	12
2.12	(a) Mecanismo de detecção (b) Circuito de acionamento [34]	13
2.13	(a) Placas utilizadas (b) Circuito de acionamento [33]	14
2.14	Diagrama de Blocos [33]	14
2.15	Robô Titan [1]	15
2.16	Robô da FVTC [23]	15
2.17	Robô Roxanne [13]	16
2.18	Robô Guitar Zero [15]	16
3.1	Célula lógica	17
3.2	Placa Altera [®] DE2-35	19
3.3	Sinal Vídeo Composto (NTSC) [14]	20
3.4	Conector RCA - Vídeo Composto	21
3.5	Conector VGA	21
3.6	Imagem representativa da tela de monitor VGA	22
3.7	Sinal VGA [3]	22
3.8	Diagrama de blocos do módulo DE2_TV [3]	23
4.1	Pixels do primeiro grupo de detecção	31
4.2	Pixels do segundo grupo de detecção	31
4.3	Pixels do terceiro grupo de detecção	32
4.4	Sequência de notas	33
4.5	Bloco limitador de frequência	33
4.6	Máquina de estados do limitador de frequência	34
4.7	Bloco <i>buffer</i>	35
4.8	Simulação do <i>buffer</i>	36

4.9	Bloco tabela de temporização	36
4.10	Fonte de alimentação	38
4.11	Subcircuito de acionamento	38
4.12	Esquemático do subcircuito de amplificação	38
4.13	Solenóide	39
4.14	Esquemático do Grybot	39
4.15	Diagrama de blocos do Grybot	40
4.16	<i>FPGA Chip Planner</i>	41
4.17	FPGA Flow Summary	41
5.1	Partitura da música <i>Through The Fire And Flames</i>	43
5.2	Controle DualShock2 e solenóides	43
5.3	Solenóide correspondente à nota verde. (a) 1 solenóide (b) 5 solenóides . .	45
5.4	Solenóide correspondente à nota vermelha. (a) 1 solenóide (b) 5 solenóides	45
5.5	Solenóide correspondente à nota amarela. (a) 1 solenóide (b) 5 solenóides .	45
5.6	Solenóide correspondente à nota azul. (a) 1 solenóide (b) 5 solenóides . . .	46
5.7	Solenóide correspondente à nota laranja. (a) 1 solenóide (b) 5 solenóides .	46

Lista de Tabelas

3.1	Especificação dos tempos de sincronismo horizontal [3]	22
3.2	Especificação dos tempos de sincronismo vertical [3]	23
5.1	Número de solenoides vs Tensão	44
5.2	Quantidade de solenoides vs Corrente	44
5.3	Posições dos pixels do grupo de detecção de notas em geral	47
5.4	Posições dos pixels para detecção de rastros (“notas contínuas”)	47
5.5	Posições dos pixels utilizados para medida da temporização	47
5.6	Intervalos de valores RGB para detecção de notas pela cor	48
5.7	Cores centrais definidas para esferas de detecção de cor	48
5.8	Intervalos de valores RGB para detecção de notas pela cor para medidas de temporização	49
5.9	Relação entre velocidades e níveis de dificuldades	49
5.10	Relação entre intervalo de tempo e velocidade	49
5.11	Tempos de espera do <i>Buffer</i>	50
5.12	Tempos do limitador de frequência	50
5.13	Desempenho obtido no nível <i>Easy</i>	51
5.14	Desempenho obtido no nível <i>Medium</i>	52
5.15	Desempenho obtido no nível <i>Hard</i>	54
5.16	Desempenho obtido no nível <i>Expert</i>	56
5.17	Desempenho na música <i>Through The Fire And Flames</i> modo <i>Training</i>	58
5.18	Grybot vs Slashbot [35, 36]	59
5.19	Grybot vs Cythbot [10, 9, 11]	59
5.20	Grybot vs Titan [20]	60
5.21	Grybot vs eGOR [25, 26]	60
5.22	Grybot vs DeepNote [6, 5]	61
5.23	Grybot vs AGH1000 [24]	61

Capítulo 1

Introdução

Em 1958, o físico Willy Higinbotham criou nos Estados Unidos o primeiro jogo para computador, chamado *Tennis for two* [4]. Era exibido na tela de um osciloscópio e representava uma simulação bem simplificada do esporte. Em 1971 foi lançado o primeiro fliperama da história chamado *Computer Space* [4], criado para suportar uma versão mais nova de *SpaceWar!* desenvolvida por Nolan Bushnell. Em 1972, Raph Baer lançou o primeiro console [4], chamado *Odyssey 100*. Desde então surgiram diversas gerações de videogames, o mercado de jogos eletrônicos sofreu uma rápida expansão e esses se fizeram mais presentes no cotidiano das pessoas.

Em razão desse crescimento, os jogos eletrônicos conquistaram espaço no meio acadêmico e se tornaram parte de pesquisas nas áreas de Inteligência Artificial (IA) e Robótica. Muitas aplicações foram encontradas no âmbito da educação, saúde, economia, engenharia, e outras. A pesquisa *Rehabilitation Robotics and Serious Games: An Initial Architecture for Simultaneous Players* [12], por exemplo, apresenta uma proposta para a arquitetura de um sistema de tratamento e reabilitação com robôs, que utiliza jogos eletrônicos para motivar e prender atenção dos pacientes, permitindo a participação simultaneamente de usuários na realização de exercícios físicos de tratamento.

A criação de algoritmos eficientes de solução de jogos representa um desafio. A criação de máquinas capazes de atingir desempenhos próximos aos que já foram alcançados por jogadores especialistas é uma meta difícil e remete a significativos avanços na área da robótica.

1.1 Contextualização

Guitar Hero^{®3} é um jogo de gênero musical desenvolvido pela *NeverSoft* inicialmente para as plataformas *PlayStation*^{®2} (PS2) e *Xbox 360*[™], que se tornou mundialmente popular entre 2005 e 2010. O jogador simula tocar uma guitarra elétrica utilizando o próprio controle do videogame ou um controlador específico, com formato do instrumento. A tela do jogo mostra o braço de uma guitarra por onde “notas musicais”, em formato de discos coloridos, passam continuamente de acordo com o ritmo e melodia da música, conforme mostra a Figura 1.1. O objetivo do jogador é tocar a música pressionando os botões do controle no momento exato em que os discos atravessam uma região específica do braço da guitarra.



Figura 1.1: Tela do jogo *Guitar Hero*^{®3}

O jogo *Guitar Hero* possui 6 sequências oficiais: *Guitar Hero II*, *Guitar Hero Encore: Rocks the 80s*, *Guitar Hero III: Legends of Rock*, *Guitar Hero: On Tour*, *Guitar Hero: Aerosmith*, *Guitar Hero World Tour*. Além de centenas de versões personalizadas criadas baseadas nas engines dos jogos originais. No Brasil foram criadas diversas versões com músicas nacionais, como *Guitar Hero: Legião Urbana*, *Guitar Hero: Gospel*, *Guitar Hero: Aviões do Forró*, *Guitar Hero: Chimbinha* entre outros.

O jogo *Guitar Hero III: Legends of Rocks (GH3)*, lançado em 2007, foi escolhido para ser usado durante o desenvolvimento deste trabalho. O GH3 possui um repertório de 70 de músicas no jogo, 45 comuns e 25 músicas bônus, cuja velocidade e quantidade de notas que aparecem na tela podem aumentar ou diminuir de acordo com a trilha escolhida e com o nível de dificuldade selecionado.

1.2 Definição do Problema

O jogo exige dos usuários muito treinamento, técnica e experiência. De forma geral, completar as músicas do repertório no nível *Expert* é uma meta difícil de ser atingida pelos jogadores. Como concluir todas as músicas do GH3 em todos os níveis de dificuldade com mais de 95% de acertos?

1.3 Hipótese

A solução do problema pode ser obtida utilizando técnicas de processamento de sinais

de vídeo sobre FPGA para comandar um sistema eletromecânico de acionamento do controle do videogame.

1.4 Objetivos

Desenvolver um robô capaz de jogar GH3 pelo controle *DualShock* e completar todas as músicas em cada nível de dificuldade com no mínimo 95% de acertos. Promover análise do sinal de vídeo, detecção das notas e desenvolver circuitos de controle para os atuadores.

1.5 Organização do Trabalho

Esta monografia foi organizada em seis capítulos: o Capítulo 1, a Introdução. No Capítulo 2 é feita uma revisão bibliográfica a respeito do tema tratado, são mencionados trabalhos relacionados a automação de jogos e, em específico, projetos de automação do jogo *Guitar Hero*[®] (GH). No Capítulo 3 encontra-se a fundamentação teórica necessária para a compreensão da construção e funcionamento do protótipo. No Capítulo 4 é apresentada uma descrição do problema enfrentado e da proposta de solução para tal. No Capítulo 5 são apresentados os resultados obtidos pela implementação do sistema. Por fim, no Capítulo 6 são apresentadas as conclusões e as propostas para trabalhos futuros.

Capítulo 2

Revisão Bibliográfica

Em 1770 foi construída a primeira máquina capaz de jogar xadrez. Várias partidas foram disputadas pela Europa e América e durante muitos anos não se soube como era possível uma máquina jogar melhor que um humano. Era chamada “O Turco”, foi criada por Wolfgang von Kempelen, que buscava impressionar a imperatriz Maria Teresa da Áustria. A máquina, apresentada na Figura 2.1, continha uma mesa com o tabuleiro de xadrez e no seu interior havia um operador que movimentava os braços de um boneco. Essa informação só fora revelada em 1820, após 50 anos da sua criação; até então os oponentes acreditavam que estavam jogando apenas contra uma máquina.



Figura 2.1: Modelo representativo da máquina “O Turco” [37]

Desde que as primeiras máquinas foram criadas, o homem tem se dedicado ao estudo de teorias e desenvolvimento de técnicas para construção de robôs capazes de realizar atividades humanas. Em maio de 1997, mais de 200 anos após a invenção de “O Turco”, a máquina Deep Blue[®], criada pela IBM[®], venceu um grupo de partidas de xadrez contra o ex-campeão mundial, Garry Kasparov.

Pesquisas nas áreas da robótica e inteligência artificial envolvendo a criação de robôs que executam trabalhos humanos são cada dia mais frequentes e amplamente aplicadas em diversas áreas de atuação como na indústria (automobilística, aeronáutica, alimentícia, farmacêutica, siderúrgica, dentre outras), na medicina, uso doméstico, em trabalhos de difícil acesso, perigosos e que envolvem situações de risco, otimizando esses processos e

oferecendo proteção às pessoas. Os resultados obtidos com essas pesquisas representam grandes avanços do conhecimento humano.

2.1 Projetos de Automação de Jogos em Geral

Nesta seção será apresentado um trabalho realizado na área de automação de jogos que se refere à criação de um robô jogador de damas [21].

2.1.1 Robô Jogador de Damas

O projeto envolve a construção de um robô capaz de jogar damas em um espaço tridimensional contra um oponente humano. É composto pelo sistema de entrada, pela *engine* do jogo e por um sistema de saída. O sistema de entrada representa a visão do robô e é responsável pela aquisição da informação visual do tabuleiro. A *engine* determina qual ação o robô deve executar. O sistema de saída realiza o movimento determinado pela *engine*.

O jogo possui peças pretas (“time 1”) e peças vermelhas (“time 2”), são divididas em duas categorias: Peões e Damas. Uma câmera é responsável capturar a imagem e o sistema de visão por extrair as informações adquiridas pela visualização do tabuleiro. A correta interpretação das posições das peças e da identificação da categoria de cada uma, exige que cada peça do tabuleiro seja enquadrada em uma imagem com, em média, 30x30 pixel (px). O plano do tabuleiro possui 8x8 posições, portanto, para que a premissa fosse cumprida, foi necessário utilizar uma câmera que suportasse resoluções superiores a 240x240 px.

A visão é afetada pelo posicionamento da câmera e pela luz do ambiente. Considerando estes fatos, a câmera foi posicionada perpendicularmente ao tabuleiro (Figura 2.2a), diminuído a influência da perspectiva e distorções. Para evitar que o sistema tivesse que trabalhar sob diversas condições de luminosidade, foi instalada uma lâmpada (Figura 2.2b), mais forte do que a iluminação ambiente, direcionada 45 graus em relação ao tabuleiro, evitando reflexos.

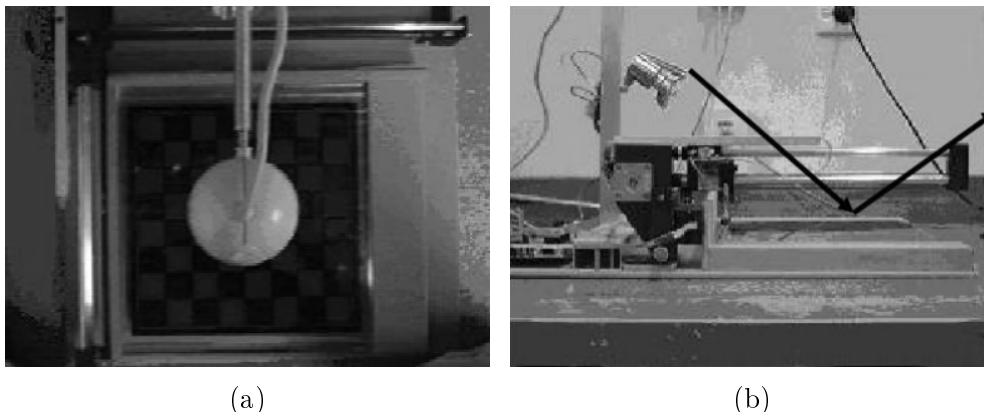


Figura 2.2: (a) Tabuleiro e câmera (b) Iluminação [21]

A identificação do time e da categoria de cada peça depende muito do contraste na imagem obtida pela câmera. O que distingue as peças de um time em relação ao outro

time são as cores, vermelho brilhoso e preto fosco. Um círculo foi pintado sobre cada peão e o símbolo de uma coroa sobre cada rei, conforme apresentado na Figura 2.3; símbolos brancos são pintados sobre peças pretas e pretos sobre peças vermelhas, obtendo um maior contraste nas cores da imagem e, conseqüentemente, facilitando a identificação.

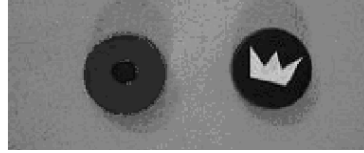


Figura 2.3: Peças peão vermelho e rei preto [21]

A função da *engine* do jogo é executar um “pulo” permitido se estiver disponível ou executar um movimento padrão se estiver disponível. O robô foi programado para executar um pulo randômico que esteja disponível ou um movimento randômico caso um pulo não seja possível.

A função do sistema de saída é realizar o movimento determinado pela *engine* do jogo, bem como ser capaz de movimentar o braço do robô nos eixos X,Y e Z em uma ordem específica e para uma posição determinada.

2.2 Projetos de Automação do Jogo *Guitar Hero*

Nesta seção serão apresentados projetos específicos de automação do jogo *Guitar Hero* que foram desenvolvidos e apresentados por estudantes, cientistas e engenheiros. A intenção aqui não é aprofundar nos detalhes de implementação, porém expor o estado da arte do tema abordado nesta monografia.

2.2.1 HeroNoid

HeroNoid é o nome do robô jogador automático de *Guitar Hero* criado em 2007 por Rafael Mizrahi e Tal Chalozin, membros de *GarageGeeks* [27]. A Figura 2.4 é uma foto desse robô.



Figura 2.4: Robô HeroNoid [27]

HeroNoid pode ser entendido por dois componentes principais, o “cérebro” e o “corpo”. O cérebro é responsável pela detecção, análise dos sinais de vídeo e geração dos comandos para o corpo. O corpo realiza a leitura dos comandos e controla a movimentação dos “dedos”. Os comandos gerados pelo cérebro são enviados via *Transmission Control Protocol* (TCP) ao corpo, neste ponto são traduzidos para um código apropriado e enviados através de um conector paralelo para a placa principal de controle do corpo. A placa recebe os comandos provenientes do computador e envia ao mecanismo de movimentação dos dedos.

Um indivíduo com plena capacidade motora não apresenta dificuldades para pressionar as teclas do controle em formato de guitarra, no entanto, essa não é uma tarefa simples para um robô. Ele precisa de uma quantidade considerável de energia para conseguir manusear as teclas. A energia é fornecida pela porta paralela com uma tensão de no máximo 3.5V. Os solenoides existentes nos dedos do robô requerem 7V e em torno de 800 mA, portanto se fez necessário o uso de um amplificador.

Uma das preocupações durante a construção deste robô foi com sua aparência. Ele se assemelha ao ser humano e suas “mãos” foram criadas para serem o mais parecidas possível as de uma pessoa. O corpo do android é um manequim humano que foi adaptado para comportar as mãos eletrônicas na posição correta. A mão esquerda utiliza os cinco dedos para realizar a captura das notas. A mão direita movimenta um dos dedos para tocar a nota, enquanto os outros 4 ficam parados. A estrutura é construída em alumínio e em cada dedo é ligado um fio acoplado a um solenoide. Em conjunto, este sistema fica responsável pela interação mecânica com o videogame.

Segundo o grupo criador do robô, diversas abordagens poderiam ser aplicadas para a detecção de objetos em movimento, entretanto o resultado obtido pela implementação da maioria delas não seria suficientemente rápido para o propósito desejado. A melhor forma de implementação encontrada foi definir pequenas regiões de captura de notas em formato de trapézio (Figura 2.5a), analisando apenas estas pequenas porções.



Figura 2.5: (a) Região de captura das notas (b) Representação binária da imagem [27]

As cores das notas são mais claras do que as cores do restante da imagem. Partindo dessa premissa, os autores optaram por representar os pixels internos aos trapézios (área de detecção) no espaço de cores HSV (*Hue, Saturation, Value*) ou HSB (*Hue, Saturation, Brightness*). Este modelo é uma transformada não linear da representação RGB cujos componentes são: tonalidade, saturação e brilho. Foi definido um limiar para o

valor de brilho, assim, obteve-se uma representação binária dos pixels dentro do trapézio, como pode ser visto na Figura 2.5b. O trapézio estando preenchido por uma quantidade suficiente de pixels brancos indica a provável passagem de uma nota.

A saída de vídeo composto do videogame foi conectada a um dispositivo de interface no computador. O dispositivo captura os quadros (*frames*) do vídeo como imagens e as envia para o programa de processamento de imagens que detecta as notas. Para tocar uma nota o jogador deve segurar pressionado o botão do controle correspondente à cor da nota e movimentar a “barra de dedilhar” da guitarra ao mesmo tempo. O reconhecimento dos botões pelo PS2 demora em torno de 100 ms, somando este valor com o tempo que o solenoide leva para se movimentar para cima e para baixo, descobriu-se que não seria possível enviar o sinal para pressionar os botões do controle e o comando para “dedilhar” ao mesmo tempo.

O protocolo utilizado foi dividido em dois grupos de ações principais: pressionamento de um ou mais botões correspondentes às cores das notas e retirada de uma nota da fila. Inicialmente a barra de dedilhar é colocada para cima e os sinais de detecção relacionados a cada cor de nota são enviados a uma fila FIFO (*First In First Out*). Após certo tempo, ocorre a retirada da nota, neste momento o sinal é enviado à guitarra juntamente com o comando de dedilhar para baixo.

O primeiro grupo de ações ocorre na região de detecção, enquanto o segundo grupo ocorre na região onde a nota é tocada. Os ajustes de tempo não são realizados automaticamente de acordo com as batidas por minuto da música (BPM). Os autores do projeto citaram que esta questão poderia ser implementada em trabalhos futuros.

2.2.2 SlashBot

Em 2008, foi construído o robô SlashBot, criado por Dave Buckner, Mitchell Jefferis, Vinny LaPenna, e Michael Voth, estudantes da *Texas A&M students* [7]. A Figura 2.6 mostra a foto do robô jogador automático de GH desenvolvido pelo grupo.



Figura 2.6: Robô SlashBot [7]

Este robô também foi dividido em dois componentes principais: o “cérebro” e o “corpo”. O cérebro é formado pelo sistema PXITM e pelo código escrito no LabVIEWTM, que analisa o sinal de vídeo. O corpo é representado por um circuito discreto que amplifica o sinal digital e que aciona os solenoides utilizados para pressionar os botões do controle do

jogo.

O sistema consiste em três diferentes módulos:

- *real-world interface* (interface com o mundo real).
- *sensing & computing unit* (unidade de detecção e computação).
- *actuation component* (componente de atuação).

A interface com o mundo real é responsável pela produção do sinal de vídeo e fornecimento de uma interface de interação com o jogo, seus componentes são:

- Sistema do PS2.
- Disco do jogo GH.
- Guitarra personalizada para o jogo.
- Monitor.

A unidade de detecção e computação possui um sistema projetado para transformar o sinal de vídeo de entrada analógico em digital. A unidade analisa o sinal e envia comandos de controle apropriados ao componente de atuação. O processamento é feito pelo sistema *National InstrumentTM (NI) PXI* mostrado na Figura 2.7, seus componentes são:

- *NI PXI-1031 System Chassis* (1).
- *NI PXI-8106 Controller* (2).
- *NI PXI-7833R Reconfigurable I/O FPGA* (3).
- *NI PXI-5114 High Speed Digitizer* (4).
- *NI SCB-68 I/O Connector Block* (5).

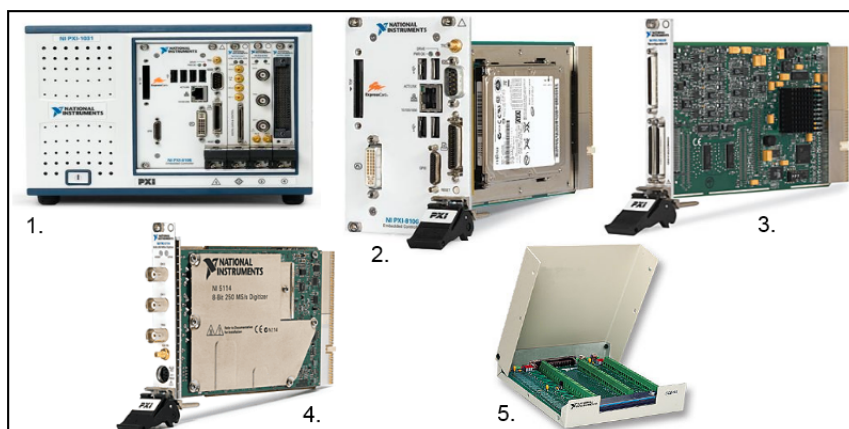


Figura 2.7: Sistema PXI [18]

O componente de atuação representa a parte robótica do projeto. É composto por sete solenoides: cinco são posicionados para acionar o botão do controle correspondente

à cor da nota (Figura 2.8a), um foi posicionado para acionar a “barra de dedilhar”, e um foi posicionado para acionar o botão de “select”, que dispara o *Star Power* (“Especial”); Circuito analógico de amplificação (Figura 2.8b); e montagem em rack (Figura 2.6).

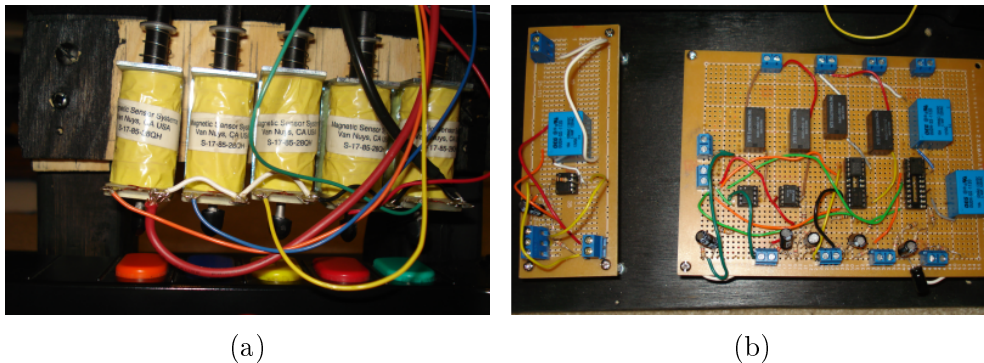


Figura 2.8: (a). Solenoides (b). Circuito analógico de amplificação [7]

O funcionamento básico do robô se inicia com a entrada do sinal de vídeo no sistema *NI PXI*, em seguida ocorre a análise de parte de vídeo ativo do sinal, detecção da nota a ser tocada e por fim é enviado um sinal apropriado ao robô para acertar a nota.

As notas que aparecem na tela possuem um brilho mais intenso comparado ao dos demais objetos, o brilho de um pixel é definido pela voltagem 0 V (preto) e 1 V (branco). A análise preto e branco da imagem permite identificar os discos com maior facilidade. O sinal em forma de onda de alguns pixels são monitorados utilizando *NI LabVIEW*. Quando uma nota atravessa o pixel monitorado ocorre um pico tensão do sinal, isso indica que ela deve ser tocada pelo robô. Quando uma detecção ocorre o sinal entra para uma fila. Após certo tempo de atraso determinado pela velocidade da música é enviado um sinal de acionamento para os solenoides.

2.2.3 CythBot

Cyth Systems construiu, em 2008, um robô jogador automático de GH chamado CythBot [16]. Uma foto do robô pode ser vista na Figura 2.9.



Figura 2.9: Robô CythBot [16]

O Cythbot não é humanoide. O robô representa uma coleção de componentes integrados capaz de detectar as notas que atravessam a tela do jogo e de responder a esse estímulo, acionando os botões do controle por meio de atuadores pneumáticos.

O sistema é composto por:

- Câmera *Pixelink*[®].
- *Compact Video System* (CVS).
- *National Instruments*[™] - *Configurable Signal Conditioning Enclosure* (CA-1000).
- Atuadores pneumáticos.
- Videogame *Wii*[™].
- Controlador *Wii*[™] (Formato de guitarra).
- Computador *Macbook*
- Monitores.

A câmera *Pixelink*[®] captura a imagem do jogo e envia ao CVS, que realiza a leitura e transferência do sinal de vídeo ao CA-1000. O sinal é processado e utilizado para acionar os atuadores pneumáticos.

Segundo os autores a estrutura do robô é representada por um modelo “produtor-consumidor”, em que o produtor é composto pela câmera e aquisição, enquanto o consumidor é composto pela guitarra e os botões.

Durante o jogo, o formato das notas podem variar entre discos e estrelas. Além disso, suas cores também podem mudar caso o especial seja disparado. Portanto, a forma de captura das notas foi baseada unicamente na intensidade de luz dos pixels de cada um dos cinco grupos de notas.

O robô foi projetado utilizando tecnologias avançadas, presentes em automação de processos industriais e testes de produtos. É construído com hardware de alta velocidade e possui um sistema que analisa sinais em tempo real.

2.2.4 eGOR

eGOR (*Eletrônica God of Rock*) é o nome do robô jogador automático de *Guitar Hero* criado em 2009 por Kyle Harrison, Jacques Mainville, e Mike McGuire e mostrado na Figura 2.10.

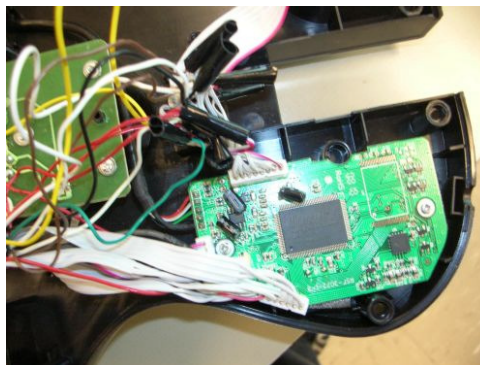


Figura 2.10: Robô eGOR [22]

Equipamentos utilizados:

- *NI Vision Builder*.
- *NI Vision Assistant*.
- *NI USB-6008 platform*.
- *2x MAX4066CPD+-ND relay chips & sockets*.
- *Unibrain Fire-ITM 1394 Camera*.

A imagem da tela do monitor é capturada pela câmera, o sinal de vídeo é enviado ao *NI Vision Assistant*, por onde é filtrado. O *NI Vision Builder* recebe o sinal, detecta os “objetos de interesse” e envia para a plataforma *NI USB-6008*. Um sinal de 5 V é levado ao relé, que fecha o circuito, permitindo que o sinal de controle seja transferido para a plataforma da guitarra. O mesmo processo é realizado para acionar os 5 botões correspondentes às notas de cada cor, para movimentar a barra de dedilhar e para disparar o especial.

A câmera utilizada adquire imagens preto e branco a uma taxa máxima de 30 quadros por segundo. Os quadros são repassados ao *Vision Assistant*. Para redução de ruídos da imagem e promover melhora na identificação de regiões importantes, a imagem passa por uma filtragem. Esse processo ocorre somente nas regiões de interesse, se tornando mais eficiente. Caixas de detecção, mostradas na Figura 2.11, foram posicionadas nos locais por onde passam as notas. O tamanho das caixas foi calibrado com muita precisão, pois afeta diretamente a correta captura de sequências rápidas de notas. Uma outra caixa de detecção foi posicionada no medidor do especial para definir o momento de acioná-lo.

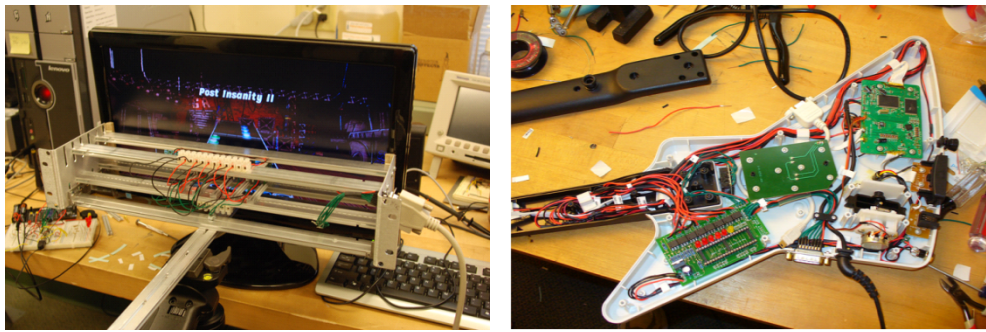


Figura 2.11: Regiões de detecção de notas [22]

2.2.5 DeepNoteTM

Estagiários da companhia de engenharia *Convolve, Inc.*, estudantes da *High School Seniors* Dr. Neil Singer, Jeremy Blum, Brandon Fischer, Zach Lynn, Alex Miller e Ben Shaffer desenvolveram um robô jogador automático de GH chamado DeepNoteTM [34] em 2008.

O robô consiste em um mecanismo de detecção (Figura 2.12a) e um circuito de acionamento (Figura 2.12b). O sinal de entrada é gerado pela detecção da luz capturada por fotodiodos. Para cada traste da guitarra foi instalado um sensor contendo um fotodiodo apontado para o local que passam as notas e um LED vermelho que ascende quando uma nota passa. O campo de visão dos fotodiodos é pequeno e seu tempo de resposta é curto (8 ns), essas características os tornam ideais para garantir precisão e análise rápida do sinal.



(a)

(b)

Figura 2.12: (a) Mecanismo de detecção (b) Circuito de acionamento [34]

O que determina a detecção de um disco são os picos de tensão, que ocorrem devido a intensidade luminosa das notas ser maior do que o resto do cenário. A tensão gerada pelos fotodiodos é muito pequena. Um circuito é responsável por amplificar esse sinal, remover ruídos e ajustar a histerese. O resultado é uma onda quadrada digital em que 3 V indica “nota” e 0 V indica “não nota”.

Os sensores ficam presos a um rack móvel e ajustável a diferentes tamanhos de televisores. O rack e a guitarra são equipados com uma porta padrão D-sub de 15 pinos, um cabo D-sub conecta os dois aparelhos. Dos quinze pinos do cabo, cinco são utilizados para transmitir os sinais dos sensores, um pino é utilizado para transmitir dados adicionais e dois outros são utilizados para fornecer o *power* e o *ground*. Os sinais digitais são enviados a partir do rack diretamente para o chip *Parallax* na guitarra. O chip *Parallax*, montado em outro PCB, apresenta pontos de saída para as funções da guitarra via relés de estado sólido.

O “cérebro” do robô é o *Parallax PropellerTM Propstick*, chip de oito núcleos que possui um sinal de relógio de 80 MHz. Nele são processadas as informações através de um programa na linguagem de alto nível *SpinTM* (linguagem própria do hardware) ou assembly.

O código implementado no chip realiza as seguintes funções principais:

1. Análise da entrada, filtragem por software, aplicação dos tempos de atraso apropriados, envio do sinal de saída para pressionar os botões da guitarra.
2. Checagem do estado dos pinos do interruptor de tempo de atraso, para que os valores possam ser ajustados de acordo com a posição do disco.
3. Verificação do momento certo de acionar o especial.

2.2.6 AGH1000

AGH1000 é o nome do robô jogador automático de GH criado em 2008 por Michael Seedman (Figuras 2.13a e 2.13b) [33].

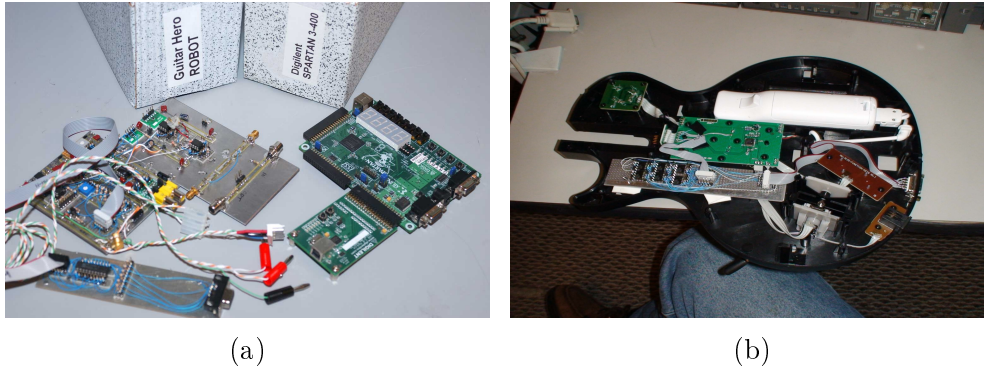


Figura 2.13: (a) Placas utilizadas (b) Circuito de acionamento [33]

O sistema é composto pelo Wii, pelo Monitor e quatro módulos (Figura 2.14):

- *Analog Processing Board* (Placa de Processamento Analógico).
- *Digital Processing Board* (Placa de Processamento Digital).
- *Driver Board* (Praca Driver).
- *Opto-Isolator Board* (Placa Opto-Isoladora).

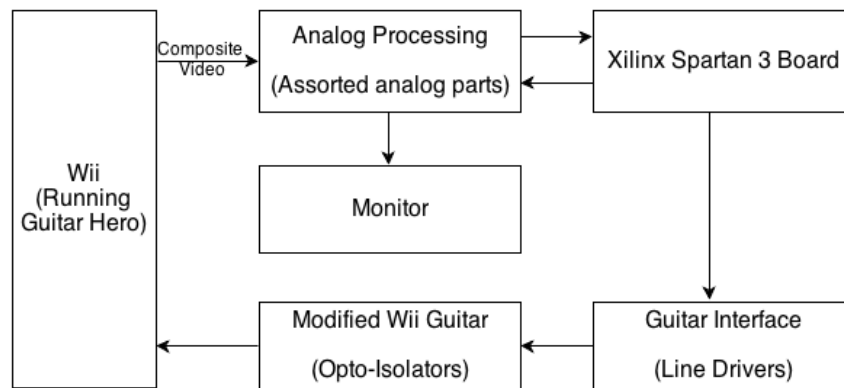


Figura 2.14: Diagrama de Blocos [33]

Conforme mostrado no diagrama de blocos da Figura 2.14, a Placa de Processamento Analógico é responsável por separar o sinal vídeo composto e gerar os sinais digitais *ODD/EVEN*, *VSYNC* e *BURST*; gerar um sinal digital se a tensão do sinal for maior que um nível chamado *SLICE*; adicionar um pequeno acréscimo de tensão ao sinal vídeo composto que leva o monitor a destacar uma região da tela.

A Placa de Processamento Digital é conectada através de um cabo fita de 10 pinos. É responsável por converter os sinais digitais da Placa de Processamento Analógico para

a Placa Driver. Seu componente principal é a FPGA, que carrega o circuito gerado pelo ISE® WebPACK™ a partir de um código em VHDL.

A Placa Driver é simplesmente um circuito integrado octal *buffer* 74LS244 usado para adequar os sinais que seguem até o cabo do opto-isolador. Cada saída aciona um opto-isolador correspondente na Placa Opto-Isoladora.

A Placa Opto-Isoladora é montada dentro da “guitarra”. É responsável por, eletronicamente, pressionar os botões, isto é, gerar os sinais elétricos que irão ser diretamente introduzidos na guitarra. Existem seis opto-isoladores nesta placa, um para cada cor (verde, vermelho, amarelo, azul e laranja), e um para ativar a barra de dedilhar.

2.2.7 Outros

Titan, mostrado na Figura 2.15, é um robô jogador automático de GH criado em 2009 por *Team Titanium FRC 1986* [1]. Ele utiliza uma *webcam* em conjunto com implementação de programação avançada para tocar as músicas do jogo.

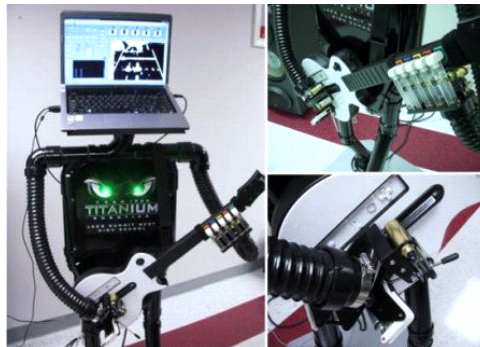


Figura 2.15: Robô Titan [1]

Em 2009, os estudantes Mike Teigen e Chee Lor, do *Fox Valley Technical College* desenvolveram também um robô jogador automático de GH [23] (Figura 2.16). Não foram encontrados registros sobre a construção e funcionamento desse robô.



Figura 2.16: Robô da FVTC [23]

Roxanne é o nome do robô jogador automático de GH criado em 2009, por Pete Nikrin [13] (Figura 2.17). O robô utiliza a câmera *Banner vision sensor* para ler a imagem

do jogo GH a partir do monitor. A detecção de uma nota ocorre quando um ponto branco brilhante passa pelo círculo de captura. Um Controlador Lógico Programável (PLC) analisa constantemente um registrador de visão, quando uma nota é reconhecida o PLC percebe a alteração no registrador e aciona o solenoide, que ativa o movimento do “dedo” do robô para tocar a guitarra.

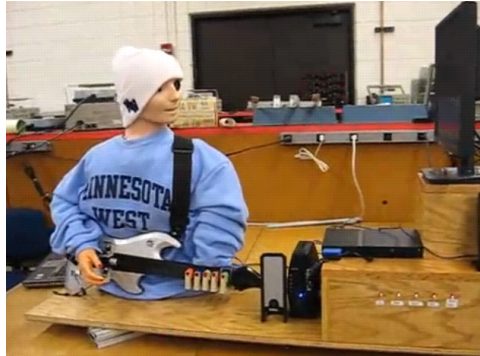


Figura 2.17: Robô Roxanne [13]

Guitar Zero é um robô jogador automático de GH criado em 2012, por Eric Falk, em Fort Wayne, IN [15] (Figura 2.18).

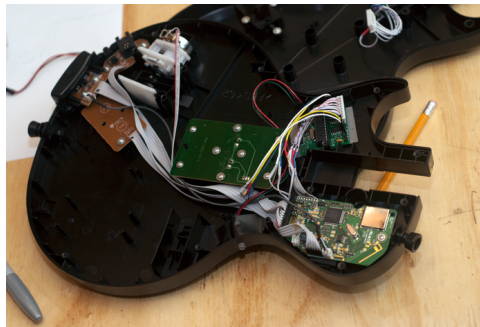


Figura 2.18: Robô Guitar Zero [15]

Diferentemente dos demais projetos apresentados, este foi construído com base num circuito programável utilizado explicitamente para gravar, salvar, compartilhar e tocar novamente as músicas do jogo *Guitar Hero*[®]/*Rock Band*[™]. Assim, não há detecção em tempo real das “notas” que passam na tela do jogo e sim ações pré-programadas para as músicas específicas, perdendo a capacidade de tocar outras músicas não carregadas.

Capítulo 3

Processamento de Sinais sobre FPGA

Este capítulo apresenta a tecnologia utilizada para o desenvolvimento do robô Grybot. Inicialmente, serão apresentados os dispositivos FPGA. A seguir os sinais de vídeo composto e a interface VGA serão descritos.

3.1 *Field Programmable Gate Array* (FPGA)

FPGAs são chips digitais de lógica programável, através dos quais é possível sintetizar qualquer função lógica [28]. São formados por centenas ou milhares de células lógicas. Em geral, cada célula é composta por uma *lookup table* (LUT), um *flip-flop D* e um multiplexador 2-para-1, conforme mostrado na Figura 3.1.

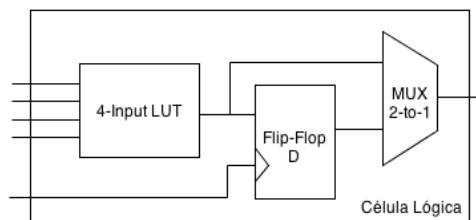


Figura 3.1: Célula lógica

Cada célula lógica pode se conectar às outras através de recursos de interconexão (fios/multiplexadores posicionados ao redor). Separada a célula não representa muito, mas o conjunto de células conectadas pode implementar diversos circuitos lógicos.

Os projetos sobre FPGA podem ser realizados de dois modos: a partir de circuitos esquemáticos ou escrita de códigos em alguma linguagem de descrição de hardware (*Hardware Description Language* - HDL). A abordagem que utiliza o desenho esquemático é interessante pois facilita a leitura, dando uma compreensão rápida do projeto como um todo. Contudo pode dificultar a manutenção de projetos muito grandes. O formato dos arquivos esquemáticos diferem de um fabricante para outro, além de ser mais simples parametrizar arquivos de descrição de hardware do que desenhos esquemáticos [29].

VHDL e Verilog [30] são linguagens HDL que se tornaram muito populares, atualmente essas são as principais linguagens suportadas pelos fornecedores de FPGA. Geralmente as HDL possibilitam três níveis de abstração [8]:

1. **Comportamental:** Permite a descrição de circuitos através de loops e processos, e possibilita compor equações que envolvem multiplicações e somas.
2. **Funcional:** Permite a descrição do circuito em termos de lógica combinacional (*if, then, else*) e booleana. Engloba também a representação do circuito no nível de registros de transferências (*Register Transfer Level - RTL*), que consiste, basicamente, em uma representação de registradores interligados por lógica computacional.
3. **Estrutural:** Representação semelhante a um *netlist* de portas lógicas ou de *switches*.

Simplificadamente, *netlist* é uma listagem de componentes do circuito e de como estes componentes estão interconectados, incluindo os nomes dos pinos de I/O do chip FPGA utilizados pelo circuito. A descrição do circuito no *netlist* é dependente do fabricante e da família do dispositivo empregado, os componentes escritos na descrição são provenientes de bibliotecas específicas do fabricante [8].

Para que o circuito seja montado na FPGA é preciso realizar a Síntese e o “Posicionamento & Roteamento”. A Síntese é responsável por traduzir o projeto (HDL ou esquemático) e criar o *netlist*. O “Posicionamento & Roteamento” descreve o processo em que os elementos do *netlist* são fisicamente alocados e mapeados aos recursos físicos do FPGA, criando assim um arquivo binário que pode ser carregado para o chip FPGA.

Atualmente Xilinx[®] e Altera[®] são os dois principais fabricantes de FPGA, suas vendas representam mais de 50% das vendas mundiais de FPGA. Altera[®] oferece FPGAs voltadas para diversos mercados (automotivo, industrial, médico, militar, testes, e outros) e aplicações (processamento de sinais digitais, processamento embarcado, prototipação ASIC, e outros) [2]. Os FPGAs de baixo custo são os da série Cyclone[®], próprios aos projetos mais simples. A Série Arria[®] fornece um equilíbrio entre performance, potência e preço, ela disponibiliza um rico conjunto de recursos de funções básicas (memória, lógica e processadores de sinal digital) e integridade superior nos dispositivos. A Série Stratix[®] possui a maior largura de banda e a maior densidade disponibilizada pelo fabricante, elas são ideais a serem utilizadas em aplicações finais. Dentro de cada serie os FPGAs são separados em famílias de acordo com suas características específicas.

Grande parte da implementação do projeto foi realizada sobre o kit de desenvolvimento DE2-35 fabricado pela Terasic (Figura 3.2). Este kit consiste em uma plataforma de desenvolvimento que permite controlar memórias, chaves, leds, botões, displays, além de diversos outros dispositivos de entrada e saída a partir do chip Altera[®] Cyclone[®] II 2C35 FPGA integrado à placa.

Os seguintes dispositivos já estão disponíveis na Placa:

1. Conector de alimentação 9V DC.
2. Porta *USB Blaster*, para comunicação com o computador.
3. *TV Decoder (NTSC/PAL)*.
4. Oscilador 27MHz.
5. Entrada de vídeo (Vídeo Composto).

6. Porta de vídeo VGA.
7. VGA DAC 10bits.
8. Oscilador 50MHz.
9. Conectores de expansão.
10. FPGA Cyclone[®] II EP2C35F672C6.
11. Memória SDRAM 8Mbyte.
12. Memória SRAM 512Kbyte.
13. LEDs verdes.
14. Chaves de pressão.
15. Chaves de seleção.
16. LEDs vermelhos.
17. *Displays* de 7 segmentos.
18. Chave de seleção modos Programação/Gravação.
19. Chips de controle do *USB Blaster*.
20. Dispositivo Altera[®] de configuração EPCS16.
21. Chave de seleção Liga/Desliga.

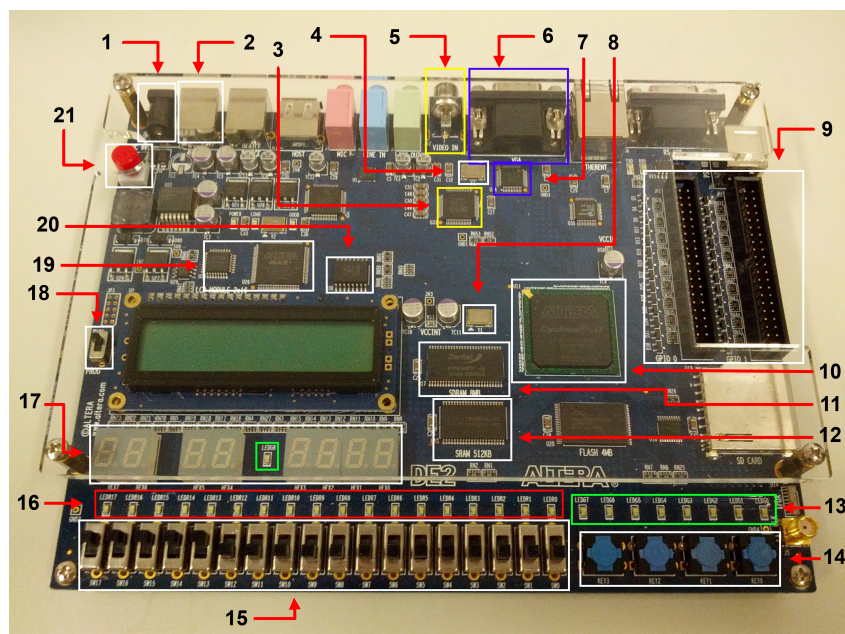


Figura 3.2: Placa Altera[®] DE2-35

A seguir são apresentados os passos principais de utilização de FPGAs:

1. Criação de um modelo esquemático digital ou da descrição textual do circuito que implementa a função lógica desejada, usando uma ferramenta de desenvolvimento fornecida pelo fabricante; no nosso caso o Quartus II.
2. Compilação da função lógica através do programa disponibilizado pelo fabricante do FPGA.
3. Conexão do FPGA através de um computador e carregamento do arquivo binário gerado pelo passo 2, no FPGA.
4. FPGA executa a função lógica especificada.

3.2 Vídeo Composto

Vídeo Composto é um sinal de vídeo analógico encontrado nos formatos padrões NTSC, PAL e SECAM. Conforme apresentado na Figura 3.3, combina em um único fio os pulsos de sincronização (linhas e quadros), os sinais de cores, e outras informações necessárias para formação da imagem.

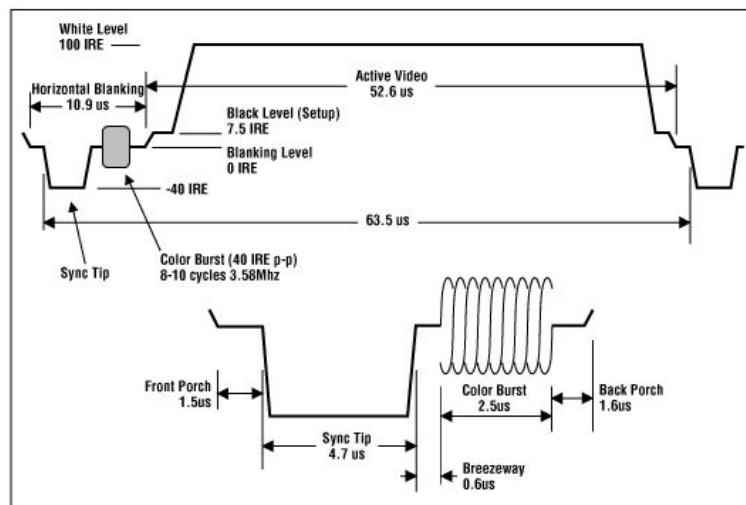


Figura 3.3: Sinal Vídeo Composto (NTSC) [14]

Este tipo de sinal possui duas componentes: crominância e luminância. A primeira se refere ao valor das cores, enquanto a segunda descreve a intensidade luminosa.

O nome do cabo utilizado para transferir o sinal vídeo composto é RCA, cujos conectores normalmente são encontrados na cor amarela (Figura 3.4).

O fato das diversas componentes do sinal estarem misturadas faz com que o sinal de vídeo resultante seja de qualidade inferior a outros padrões em que as componentes são transferidas separadamente.



Figura 3.4: Conector RCA - Vídeo Composto

3.3 *Video Graphics Array (VGA)*

Video Graphics Array (VGA) é um padrão de vídeo introduzido pela IBM[®] em 1987. O conector utilizado para ligar os dispositivos eletrônicos que usam este padrão é chamado Conector D-subminiature de 15 pinos. Dá-se esse nome porque os conectores deste tipo são normalmente circundados por um envoltório metálico em formato de D, que protege contra interferência eletromagnética e garante um encaixe correto.



Figura 3.5: Conector VGA

A imagem apresentada na tela é gerada por uma combinação de pixels que continuamente se acendem e se apagam. A frequência que isso ocorre é muito alta, portanto essa variação não é perceptível ao olho humano. Cada pixel possui três componentes de cores: vermelho, verde e azul; o monitor varre a tela de 640x480px 60 vezes por segundo alterando essas componentes e isso resulta na imagem gerada na tela.

A varredura é iniciada na primeira linha e primeira coluna da tela, que fica no canto superior esquerdo do monitor, e se move para a direita até encontrar a última coluna. Quando a varredura encontra a última coluna, a leitura recomeça no início da linha seguinte. Quando chega ao último pixel do canto inferior direito da tela, a varredura retorna à posição de inicial e recomeça o processo. Durante cada retrocesso horizontal e vertical todos os pixels são apagados; este processo é representado na Figura 3.6.

O monitor VGA é controlado por cinco sinais: as componentes vermelho, verde e azul, os sinais de sincronização horizontal e sincronização vertical. Os sinais de cores são chamados de sinais RGB e controlam a cor de cada pixel dada uma posição da tela. São sinais analógicos, representados por tensões entre 0.7V e 1V que definem a intensidade das cores. Os sinais de sincronização horizontal e vertical são responsáveis por controlar a temporização da taxa de varredura. São sinais digitais que assumem valor 0 ou 1. O sinal

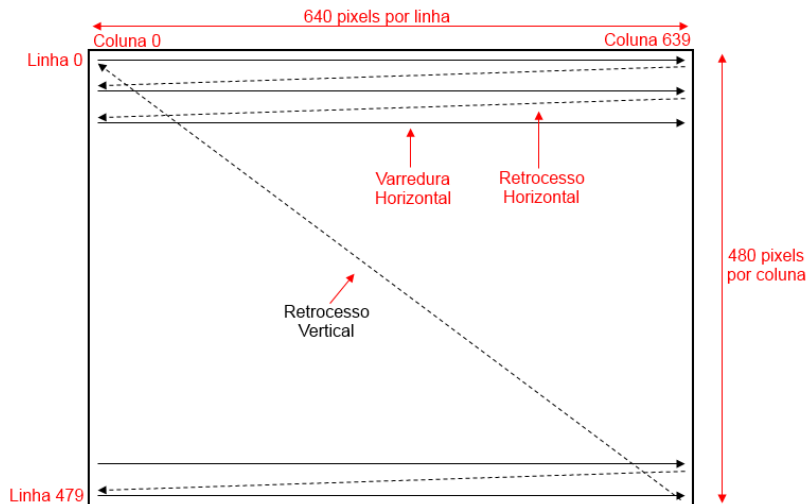


Figura 3.6: Imagem representativa da tela de monitor VGA

horizontal determina o tempo que demora para varrer uma linha e o vertical determina o tempo que leva para varrer todo o quadro (*frame*).

Conforme pode ser visto na Figura 3.7, um pulso baixo de certo período (a) é aplicado ao sinal de sincronização horizontal (*hsync*), que indica o término de uma linha e o começo da próxima. A entrada de dados RGB no monitor deve ficar em 0 V por um período de tempo chamado *back porch* (b) depois que um pulso baixo *hsync* ocorre. Após esse período ocorre o intervalo de exibição (c). Durante esse intervalo os dados RGB são conduzidos a cada pixel através da linha que está sendo exibida. No final, antes do próximo pulso baixo *hsync*, o sinal RGB deve ficar novamente em 0 V durante um período chamado *front porch* (d). O término de um frame e início do próximo respeita as mesmas características de temporização utilizadas para término e início de linhas, a diferença é que o controle é feito pelo sinal de sincronização vertical (*vsync*) e os tempos são representados pelo período necessário para uma linha em quantidade de linhas.

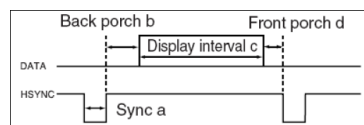


Figura 3.7: Sinal VGA [3]

A Tabela 3.1 apresenta, em μs , os tempos de sincronismo horizontal referentes ao padrão de vídeo utilizado no projeto.

Tabela 3.1: Especificação dos tempos de sincronismo horizontal [3]

	a	b	c	d
Tempo (μs)	3,8	1,9	25,4	0,6

A Tabela 3.2 apresenta os tempos de sincronismo vertical. A unidade de medida das informações da tabela é número de linhas. O tempo de cada linha corresponde ao somatório dos tempos de sincronismo horizontal: 31,7 μ s.

Tabela 3.2: Especificação dos tempos de sincronismo vertical [3]

	a	b	c	d
N ^o linhas	2	33	480	10

A placa Altera[®] DE2-35 possui um conector D-sub de 15 pinos para a saída VGA. Os sinais de sincronização são fornecidos diretamente do Cyclone[®] II FPGA, que implementa a controladora VGA, e o dispositivo ADV7123 (*triple 10bit high speed video DAC*) é responsável pela produção dos sinais analógicos (vermelho, verde, azul).

3.4 Módulo DE2 TV

Este módulo, descrito em Verilog, é disponibilizado pelo fabricante da placa DE2-35. Se trata de um circuito capaz de receber o sinal vídeo composto como entrada e de gerar a imagem através da saída VGA. Conforme mostrado na Figura 3.8, é composto de dois grandes blocos, I2C_AV_Config e TV_to_VGA. Os chips TV Decoder (ADV7181) e VGA DAC (ADV7123) controlam a entrada e saída dos sinais.

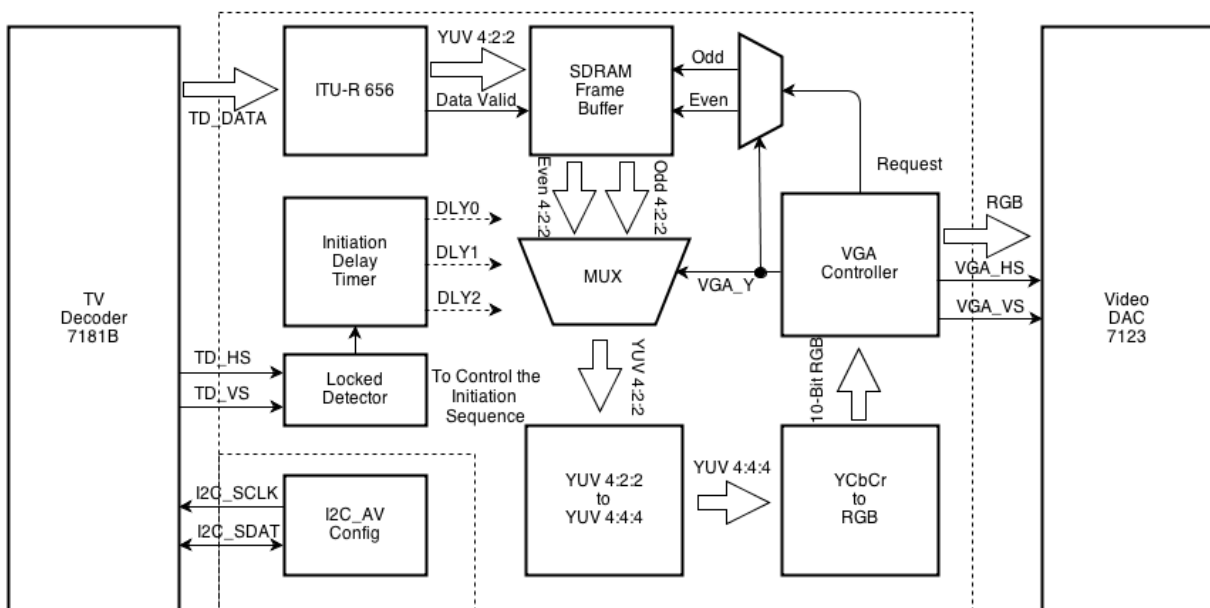


Figura 3.8: Diagrama de blocos do módulo DE2_TV [3]

Inicialmente o circuito descrito no arquivo binário é carregado para a FPGA. O chip TV Decoder configura o decodificador através do bloco I2C_AV_Config, que utiliza o protocolo I2C para se comunicar com esse chip. Após a sequência de ativação fica instável por um tempo, então o detector identifica essa instabilidade.

O sistema TV_to_VGA é composto pelos blocos:

- *ITU-R 656 Decoder.*
- *SDRAM Frame Buffer.*
- *YUV422 to YUV444.*
- *YCrCb to RGB.*
- *VGA Controller.*

O Bloco *ITUR 656 Decoder* extrai o sinal de vídeo no formato YCrCb 4:2:2 (YUV 4:2:2) do fluxo de dados enviados pelo *TV Decoder* e gera um sinal de controle que indica o período válido de envio dos dados. A controladora VGA envia uma requisição par/impar ao *SDRAM Frame Buffer* e multiplexador de seleção de campos que permite desentrelaçar o sinal de vídeo e enviar para o Bloco *YUV422_to_YUV444*. Neste ponto o sinal selecionado é convertido do formato YCrCb 4:2:2 (YUV 4:2:2) para YCrCb 4:4:4 (YUV 4:4:4). Por fim, o bloco *YCrCb_to_RGB* fica responsável por converter os dados em YCrCb para saída RGB. A controladora VGA gera os sinais de sincronismos horizontal e vertical: *VGA_HS* e *VGA_VS*. Isso permite que a imagem seja mostrada em um monitor VGA.

Capítulo 4

Modelo Robótico Proposto

Este capítulo aborda questões encontradas durante a construção do robô jogador automático de *Guitar Hero*^{®3} (GH3). Serão apresentadas as técnicas utilizadas e o funcionamento dos mecanismos desenvolvidos para atingir o objetivo inicial do projeto.

4.1 Descrição do Problema

Enquanto uma pessoa joga, diversas informações são processadas no seu cérebro sem que ela perceba. A captura da imagem é feita pelos olhos. A seguir, a imagem é analisada pelo cérebro, que é responsável por decidir qual ação será tomada, escolher o momento em que ela será realizada e enviar um comando para a parte do corpo que deve executá-la. O corpo reage ao estímulo, então o processo é finalizado.

Semelhante ao que acontece com o ser humano, estas ações deverão ocorrer também a um robô programado para substituí-lo nessa atividade. Nesta seção serão apresentadas algumas alternativas para a implementação dos mecanismos do robô responsáveis pela execução de suas tarefas.

4.1.1 Captura do sinal de vídeo

O primeiro passo para a execução das tarefas envolvidas no jogo é a captura da informação visual. O robô precisa de um sensor que lhe permita identificar os objetos do jogo. A forma mais comum é através da extração de imagens de câmeras; uma alternativa viável é a captura do sinal vídeo composto, obtido do videogame.

1. Câmera:

Classificações:

- Tipo de sensor: CCD, CMOS, Outros.
- Tipo de sinal: Analógico, Digital.
- Interface: Analógica (RS-170, NTSC, etc.), Digital (USB, Firewire, GigE, CL).
- Cor: Monocromática, Colorida (1 sensor), Colorida (3 sensores).
- Formato do sensor: Area Scan, Line Scan.
- Tipo de varredura: Progressive Scan, Interlaced Scan.

(a) Vantagens:

- A grande variedade de câmeras, permite ao desenvolvedor do projeto escolher o tipo do sensor que melhor se enquadra às necessidades e ao orçamento do seu projeto.
- As técnicas de visão computacional aplicadas ao projeto podem ser mais facilmente aproveitadas em outros trabalhos, já que a implementação não fica restrita a um padrão de vídeo específico.

(b) Desvantagens:

- O desenvolvimento de aplicações que ocorrem em tempo real e exigem processamento rápido da imagem, necessitam de câmeras altamente específicas. Dessa forma, o custo/benefício desta abordagem pode se tornar descompensador ou até inviável.
- A imagem adquirida por este meio sofre influência da iluminação do ambiente, reflexos, distorções, que podem afetar o processo de análise da imagem.

2. Sinal vídeo composto:

A imagem gerada pelo videogame é enviada a um televisor através do sinal vídeo composto. Este sinal pode ser capturado pelo robô e servir como entrada para seu sistema de visão.

(a) Vantagens:

- Não necessita de um sistema de captura do sinal a partir do monitor, o sinal é disponibilizado como saída do videogame e pode ser processado diretamente pelo robô; reduzindo, assim, o tempo de aquisição das informações.
- Não é afetado pelas características do meio em que o robô é colocado.

(b) Desvantagens:

- Por se tratar de um sinal analógico, é preciso que o robô contenha um componente capaz de transformá-lo em sinal digital antes de ser analisado.
- Análises de regiões de imagem ficam mais complexas de serem realizadas.

4.1.2 Detecção de notas

Após definir como o sinal de vídeo será capturado é preciso decidir qual método de detecção de notas será utilizado.

1. Cor

No jogo, cada nota possui uma cor (Figura 1.1). A abordagem de detecção baseada em cor utiliza o valor das componentes RGB dos pixels da imagem para detectar o momento em que os discos passam pela região de interesse.

(a) Vantagens:

- Detecção mais especializada das notas, identificando apenas objetos de cores específicas e desconsiderando possível identificação de objetos do plano de fundo.

(b) Desvantagens:

- Quando o usuário aciona o especial as cores das notas mudam, desta forma é preciso levar esta questão em consideração.

2. Formato

No jogo GH, cada nota apresenta sempre dois formatos: disco ou estrela.

(a) Vantagens:

- Permite capturar as notas sem que seja levado em consideração a sua cor.
- Permite diferenciar notas comuns de estrelas.

(b) Desvantagens:

- É necessário utilizar vários pixels da imagem para realizar a análise da forma.

3. Posição na tela

Em geral, sistemas que envolvem visão computacional utilizam áreas específicas da imagem para a detecção de objetos, as regiões de interesse (RDI). Neste jogo, há uma perspectiva aplicada sobre o braço da guitarra que dá ao usuário a impressão de que as notas aparecem do fundo da imagem e se deslocam para a parte frontal. Em consequência, quando as notas estão próximas à parte superior da tela elas são menores (ocupam menor quantidade de pixels da tela) e quando elas estão na parte inferior são maiores (ocupam maior quantidade de pixels da tela).

Para efeito de detecção das notas seria mais interessante que o método escolhido monitorasse a parte mais inferior da tela, pois é o local em que os objetos estão maiores, e portanto mais facilmente identificáveis. Entretanto, cada vez que uma nota é tocada corretamente aparece, no local, uma animação em formato de chamas (Figura 1.1) que atrapalha o processo de detecção. Sendo assim, a melhor opção é um método alternativo que será detalhado na próxima seção.

4. Janela

É preciso decidir a quantidade de pixels que serão utilizados para definir a área da detecção das notas. Uma das abordagens é a utilização de uma janela retangular de pixels.

(a) Vantagens:

- Detecção pode se tornar mais apurada, pois engloba uma quantidade maior de pixels.

(b) Desvantagens:

- Dificuldade em se delimitar exatamente o tamanho da janela, as notas possuem tamanho específicos.
- Análise se torna mais complexa, pois envolve quantidade maior de pixels.

4.1.3 Sincronização temporal

No jogo GH3, antes de tocar uma música o jogador deve selecionar um dos seguintes níveis de dificuldade:

- *Easy*
- *Medium*
- *Hard*
- *Expert*

A quantidade de notas e a velocidade em que elas aparecem variam de acordo com a opção escolhida. No modo de treinamento é possível reduzir a velocidade em que a música é tocada, as opções disponíveis são:

- *Slowest*
- *Slower*
- *Slow*
- *Full Speed* (não há redução de velocidade)

O robô deve ser capaz de sincronizar o tempo de pressionamento das teclas com a velocidade em que a música é tocada. Para tal, é preciso definir um método que estima o tempo gasto pelas notas para se deslocarem uma distância na tela. O valor medido é utilizado para configurar os parâmetros de temporização.

4.1.4 Calibração

Toda máquina precisa estar devidamente calibrada para executar de modo satisfatório as tarefas que lhe são designadas. Os ajustes do software ou do hardware possuem grande impacto no resultado final. No desenvolvimento do projeto são definidas algumas constantes numéricas que precisam ser escolhidas com muita precisão.

1. Intervalo ou Limiar RGB:

A identificação de objetos envolve detecção de cores RGB. Duas formas podem ser utilizadas neste processo: Captura de cores através de um intervalo definido ou captura através da escolha de um limiar RGB, ou seja, limitar por cores mais claras.

2. Posições de detecção na tela. O sistema de visão é responsável pela captura da imagem. Na maioria das vezes apenas uma região, ou grupo de pixels são utilizados para a análise da imagem. Assim, é preciso definir os pontos da tela que serão utilizados para esta função. A variação de posição da tela pode influenciar significativamente no resultado.

4.1.5 Interação

É certo que, de algum modo, o robô deve interagir com o sistema do jogo. Ele pode fazer isso através de estímulos mecânicos ou estímulos elétricos.

1. Estímulos Elétricos

A transmissão do comando ao sistema do jogo é realizada através de ligações elétricas do circuito do robô ao controle do videogame.

(a) Vantagens:

- Reduzido tempo de resposta.
- Baixa influência do meio em que se encontra.
- Não ocorre desgaste físico de componente.

(b) Desvantagens:

- É preciso modificar os componentes do sistema do jogo, para criar as ligações diretas.
- É considerado por alguns desenvolvedores como sendo *hack*, isto é, pode ser visto como trapaça por necessitar modificar o circuito do controle.

2. Estímulos Mecânicos

A transmissão do comando ao sistema do jogo é realizada através do pressionamento das teclas do controle do videogame. Atuadores pneumáticos e solenoides podem ser escolhidos para execução dessa tarefa.

(a) Vantagens:

- Ação semelhante aos dedos humanos.
- Não intrusivo ao sistema eletrônico do videogame.

(b) Desvantagens:

- Necessita modelagem física do dispositivo (tempo de resposta, força).
- Sofre influência do meio.
- Sofre desgaste com o tempo.

4.2 Descrição do Projeto

O robô construído neste projeto é composto por um grupo de dispositivos eletromecânicos que trabalham em conjunto para jogar automaticamente o jogo GH3. Inicialmente será explicado como o sistema robótico foi construído e, então, o seu funcionamento será detalhado.

4.2.1 Detecção de Cores

O módulo Detecção e cores é responsável pela captura e análise da imagem, deve ser capaz de identificar em tempo real todas as notas que atravessam o braço da guitarra. O sinal vídeo composto é gerado pelo PS2, enviado pelo cabo RCA amarelo e recebido pela

placa DE2-35 através do *TV Decoder* do circuito DE2 TV descrito na Seção 3.4. Então, é feita a digitalização, desentrelaçamento do sinal, conversão de padrões e por fim o sinal é enviado da controladora VGA para o *Video DAC*.

Após a última etapa desse processo o sinal de vídeo fica pronto para ser enviado a um monitor que ofereça suporte ao padrão VGA. O sinal utilizado para analisar a imagem do jogo é obtido através da interceptação do sinal de saída da controladora VGA. Neste ponto é possível obter o valor de cada componente RGB do pixel e a posição do pixel que é aceso na tela em dado momento.

Grupos de pixels em regiões específicas da tela foram definidos para serem monitorados. A controladora VGA envia a posição e o valor das componentes RGB de um pixel; quando a posição enviada coincide com a posição de um pixel que faz parte de um dos grupos de detecção, as cores são analisadas e as ações correspondentes são tomadas (enviar o sinal “detectou nota” ou “não detectou nota”).

Em geral, a implementação de sistemas de visão computacional varia bastante, no entanto algumas funções genéricas estão quase sempre presentes:

- Aquisição de imagem: imagem digital produzida por sensores.
- Pré-processamento: processamento da imagem de modo a assegurar que as condições do método aplicado são atendidas, realizando também redução de ruídos e aumento de contraste.
- Extração de Características: extração de características da imagem (bordas, cantos, formatos e texturas).
- Detecção e Segmentação: decisões são tomadas a partir de uma ou mais regiões relevantes da imagem.
- Processamento de alto nível: os dados de entrada desta etapa são pequenos, nesse ponto ocorre a classificação dos objetos detectados.

Diferentemente dos sistemas de visão computacional comuns, o sistema apresentado neste projeto não analisa a imagem completamente. A análise é realizada sobre a varredura e detecção de pixels específicos da imagem ao invés de regiões ou janelas. A aquisição da imagem é feita através do sinal vídeo composto. A intensidade das cores dos objetos a serem identificados é mais alta do que as do plano de fundo do jogo (“braço” da guitarra). Esta característica assegura que o método de detecção dos objetos possa ser aplicado. Grupos de pixels em regiões específicas da tela são utilizados para a tomada de decisões ou ações. O resultado final é um sinal “detectou nota” ou “não detectou nota” gerado de acordo com a classificação das componentes RGB dos pixels.

A imagem produzida no padrão VGA utilizado ocorre a uma taxa de 60Hz e é representada em resolução 640x480px. Isso significa que 60 quadros são apresentados por segundo; cada frame possui 480 linhas e cada linha contém 640 pixels.

Grupos de detecção:

1. Grupo de pixels para detecção de notas em geral.
2. Grupo de pixels para detecção de rastros (“notas contínuas”).
3. Grupo de pixels utilizados para temporização.

O objetivo específico do jogo é acertar o momento correto de pressionar um ou mais botões do controle correspondentes às cores das notas. São encontradas notas verdes, vermelhas, amarelas, azuis e laranjas.

Conforme mostrado na Figura 4.1, o primeiro grupo de detecção é composto de 5 pixels, um em cada traste da guitarra por onde passam as notas de cada cor. As posições dos pixels foram definidas de modo que não houvesse interferência da animação do jogo no processo de análise. Então, foram colocados logo acima da região em que as chamas geradas pelo acerto da nota são renderizadas.

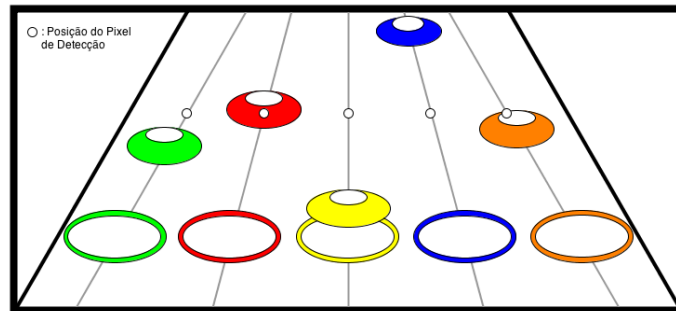


Figura 4.1: Pixels do primeiro grupo de detecção

Sempre que a controladora VGA envia o sinal de um pixel cujas posições x e y coincidem com as posições de um dos pixels do grupo, o sistema verifica se os valores das componentes enviados combinam com os valores especificados para detecção das notas que passam pelo ponto, respeitando a Equação 4.1. Caso isso ocorra, é gerado um sinal digital 1, caso contrário 0.

$$Detecta = ((R_{min} < R_i < R_{max}) \&\& (G_{min} < G_i < G_{max}) \&\& (B_{min} < B_i < B_{max})) \quad (4.1)$$

O segundo grupo de detecção, apresentado na Figura 4.2, é formado por 65 pixels. São utilizados 13 pixels por traste posicionados logo acima do local usado pelo primeiro grupo. Sua função é indicar quando o rastro de uma nota contínua está passando, a quantidade de pixels desse grupo foi definida experimentalmente e representa a quantidade mínima de pontos necessários para capturar o rastro.

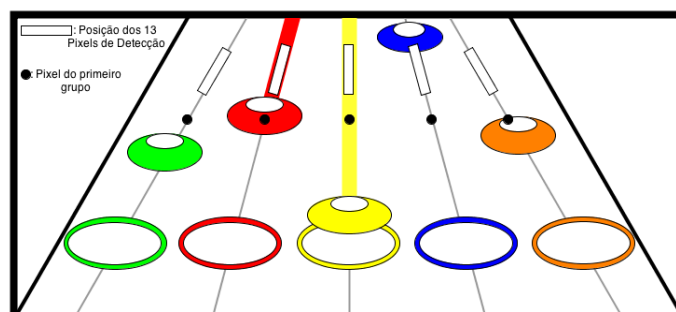


Figura 4.2: Pixels do segundo grupo de detecção

Inicialmente, foi definida uma equação de detecção para as cores verde, vermelho, amarelo, azul e laranja que corresponde às cores das notas do jogo. Então, sempre que a controladora envia os sinais de um pixel que será monitorado, o sistema verifica se os valores das componentes enviados se enquadram nos valores definidos pela equação da esfera de detecção (Equação 4.2).

$$Detecta = ((R_i - R_{base})^2 + (G_i - G_{base})^2 + (B_i - B_{base})^2) < Dist^2 \quad (4.2)$$

A cada pixel do grupo foi associado um registrador de um bit que armazena o resultado da verificação. Após passar pelo último pixel do grupo o valor contido nos registradores são somados e o valor gerado segue a um comparador que verifica se o resultado final foi maior que 11. Caso tenha sido, é gerado um sinal digital 1, caso contrário 0.

O terceiro grupo de detecção é composto por 10 pixels, 2 para cada cor de nota (Figura 4.3). Esse sistema é utilizado para contar o número bordas de descida que ocorrem a um *clock* de 27MHz enquanto as notas do jogo atravessam de um ponto da tela a outro.

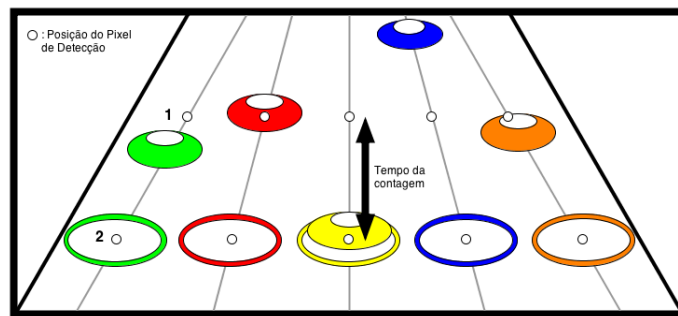


Figura 4.3: Pixels do terceiro grupo de detecção

Quando a controladora VGA envia o sinal correspondente ao primeiro pixel do grupo definido para cada cor de nota, é feita uma comparação entre o valor das componentes RGB enviadas pela controladora e o intervalo de valores definidos para detecção das notas que atravessam cada ponto (Equação 4.1). A comparação retorna 1 se os sinais do pixel monitorado corresponde ao de uma nota musical e retorna 0 se o contrário ocorrer. Caso o retorno seja 1, o circuito inicia um contador que é incrementado a cada borda de descida do oscilador 27MHz. A contagem para quando a controladora VGA envia o sinal correspondente à posição do segundo pixel do grupo definido para cada traste e cuja comparação de componentes também corresponde a de uma nota musical.

4.2.2 Temporização

O limitador de frequência e o *buffer* são dois componentes essenciais para o funcionamento do robô. Os parâmetros de entrada necessários para a realização de suas tarefas são completamente dependentes da informação temporal que indica a velocidade de operação do robô.

1. Limitador de Frequência:

O sinal de detecção gerado pelo módulo detecção de cores não é suficiente para que o robô possa atingir um bom desempenho no jogo. O principal motivo disso se baseia

no fato de que, em alguns momentos, as notas aparecem muito próximas umas das outras, conforme mostrado na Figura 4.4. Quando isso ocorre, o módulo detecção de cores não consegue separar o término de uma nota e início da outra.

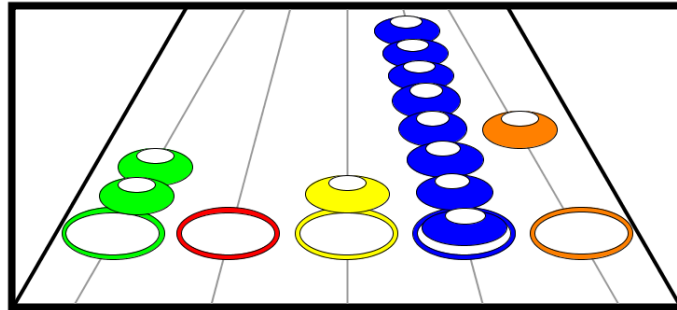


Figura 4.4: Sequência de notas

Para corrigir este tipo de problema foi criado um bloco limitador de frequência (Figura 4.5) que define o tempo que o robô deve manter a tecla do controle pressionada, e o tempo em que ela deve ficar solta.

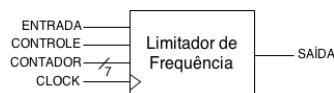


Figura 4.5: Bloco limitador de frequência

Os sinais de entrada do bloco são: CLOCK, ENTRADA, CONTADOR, CONTROLE. O sinal de saída do bloco é: SAIDA

O primeiro sinal de entrada do bloco é um sinal de relógio de 50MHz; sinal ENTRADA é proveniente do módulo detecção de cores, saída do bloco do grupo de pixels de detecção de notas em geral; o sinal CONTADOR é uma constante que define a velocidade de operação do robô, ela varia de acordo com o nível e redução de velocidade escolhido pelo jogador; o sinal CONTROLE é proveniente do módulo detecção de cores, saída do bloco do grupo de pixels de detecção de notas contínuas.

O sinal SAIDA, do limitador de frequência, é determinado por uma máquina de seis estados, apresentada na Figura 4.6. O primeiro estado é o inicial, ele é responsável por zerar o contador interno do bloco; o sinal de saída desse estado é o valor do sinal de entrada. Quando o sinal de entrada é 1, a função de transição da máquina leva ao segundo estado, caso contrário, permanece no mesmo. O segundo estado representa um período de estabilização do sinal de entrada. O contador interno é incrementado a cada subida da borda de CLOCK, permanece no mesmo estado até que um valor determinado por constante α multiplicada pelo CONTADOR seja atingido; o sinal de saída deste estado é 0. Se o sinal de entrada continua em 1 a função de transição da máquina leva ao terceiro estado, caso contrário retorna ao estado inicial; o sinal de saída do terceiro estado é 1 e sua função é incrementar o contador a cada subida

de clock, a máquina permanece neste estado até que o contador atinja o valor determinado por constante β multiplicada pelo CONTADOR, depois segue ao quarto estado. O quarto estado é responsável por zerar o contador interno; a função de transição desse estado leva apenas ao quinto estado. No quinto estado o sinal de saída é 0; o contador é incrementado, a máquina permanece nesse estado até que o valor determinado por constante γ multiplicada pelo CONTADOR seja atingido, depois retorna ao estado inicial. O CONTROLE é um sinal assíncrono com o resto da máquina; provém do módulo detecção de cores, saída do bloco do grupo de pixels de detecção de notas contínuas. Quando seu valor é 1, o sinal de saída do circuito retorna sempre 1, e o contador interno é zerado. Quando o sinal CONTROLE vai para 0 a máquina segue ao sexto estado, que representa um estado de correção de perspectiva da imagem; sua função é manter o sinal de saída em 1; o contador é incrementado até que atinja o valor da constante δ multiplicada por CONTADOR; a função de transição desse estado leva ao estado inicial.

$$c1 = \alpha * CONTADOR \quad (4.3)$$

$$c2 = \beta * CONTADOR \quad (4.4)$$

$$c3 = \gamma * CONTADOR \quad (4.5)$$

$$c4 = \delta * CONTADOR \quad (4.6)$$

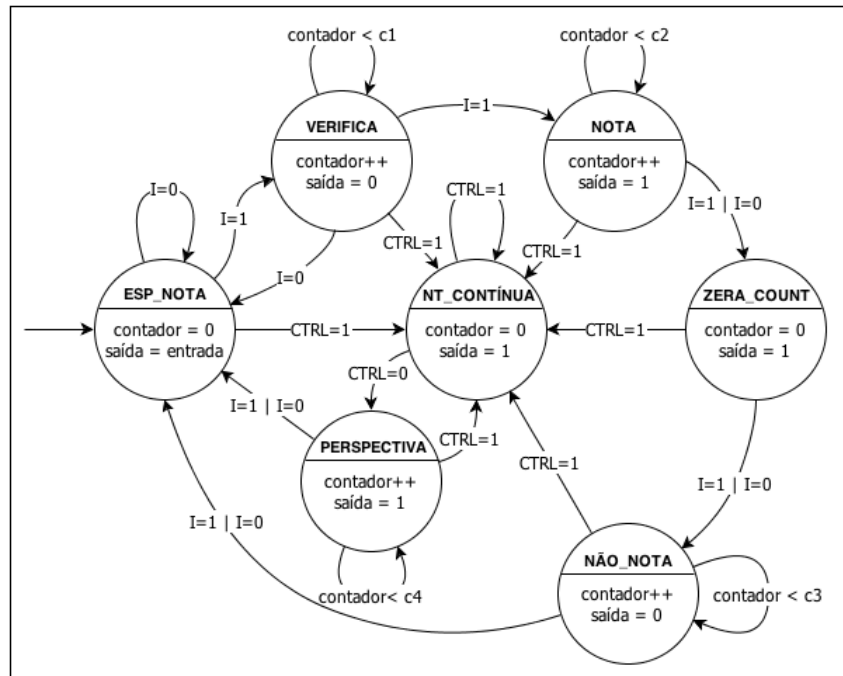


Figura 4.6: Máquina de estados do limitador de frequência

2. *Buffer*:

A região utilizada para detecção das notas não é a mesma em que as notas devem ser tocadas. O sinal gerado pelo módulo de detecção de cores precisa ser armazenado durante o tempo em que as notas levam para alcançar a posição da tela em que são tocadas.

O *buffer*, mostrado na Figura 4.7, é o bloco responsável por gerar o tempo de atraso necessário para que o sinal de detecção das notas fique sincronizado com o tempo de acionamento dos botões do controle correspondentes às notas que aparecem na tela.

Os sinais de entrada do bloco são: CLOCK, ENTRADA e CONTADOR. O sinal de saída do bloco é: SAIDA.



Figura 4.7: Bloco *buffer*

O primeiro sinal de entrada do bloco é um sinal de relógio de frequência 75Hz; o sinal ENTRADA é proveniente do bloco “limitador de frequência”; CONTADOR é uma constante que define a velocidade de operação do robô, ela é determinada da mesma forma que a constante definida no bloco limitador de frequência, porém possui valor distinto.

O bloco possui um registrador interno de 6 bits que armazena a posição da “célula” logo acima da última célula do *buffer* que contém valor. O *buffer* é formado por 64 células (posições), cada uma pode armazenar o valor de um contador que contenha no máximo 6 bits.

A função do *buffer* é atrasar o sinal ENTRADA por um período que depende do sinal CONTADOR e da frequência de CLOCK.

A estrutura criada para o *buffer* será melhor compreendida por meio da descrição do que ocorre internamente ao bloco a cada borda de descida do CLOCK, e da Figura 4.8.

1º Passo - O valor de ENTRADA está em 1; O bloco insere o valor de CONTADOR da célula indicada pelo registrador de posição; O registrador de posição é incrementado em 1.

2º Passo - O valor do contador de todas as células do *buffer* que armazenam valores maiores do que 0 são decrementados em 1.

3º Passo - O bloco testa se o valor do contador da célula[0] é igual a 0 e se o registrador de posição também é maior que zero. Se ambas as premissas forem verdadeiras, então o sinal SAIDA é alterado para 1 e cada célula do *buffer* com exceção da última, recebe o valor do contador da célula seguinte; o registrador de posição é decrementado em 1. Se a uma das premissas for falsa, o sinal SAIDA fica em 0.

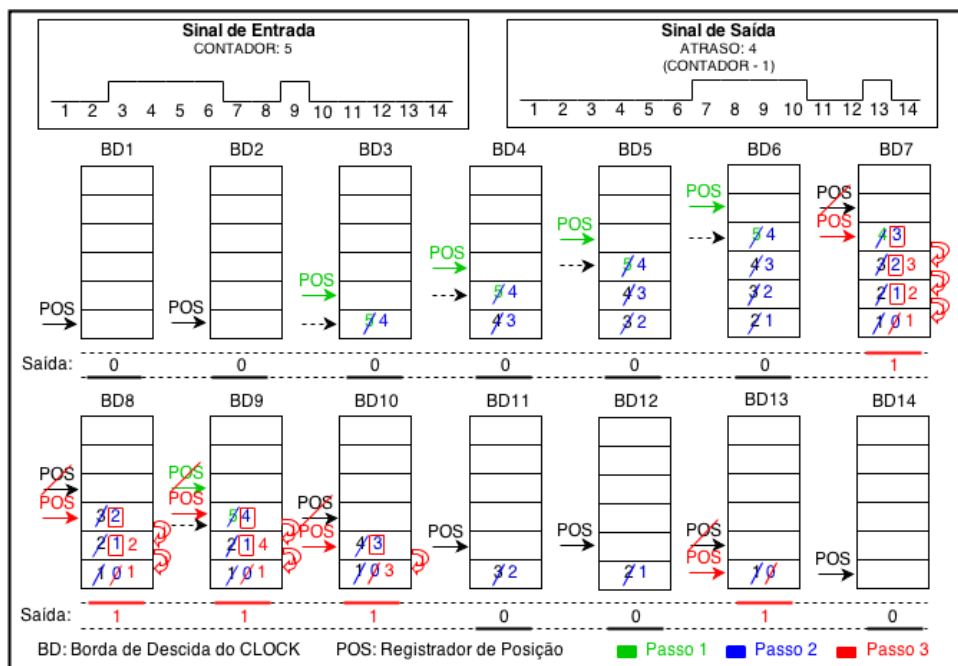


Figura 4.8: Simulação do *buffer*

3. Tabela de temporização:

A velocidade em que as notas de uma música do jogo atravessam o braço da guitarra varia de acordo com as configurações definidas pelo jogador antes do início da música. O robô desenvolvido neste projeto é capaz de identificar essa velocidade, o reconhecimento é feito com base no do valor do contador gerado pelo módulo detecção de cores a partir do terceiro grupo de pixels de detecção, descrito na subseção 4.2.1.

Tabela de temporização é o bloco responsável por gerar o contador de entrada do limitador de frequência e o do *buffer*. Os sinais de entrada do bloco são: CLOCK e CONTADOR; os sinais de saída são: CONTADOR_LF e CONTADOR_BUFF.

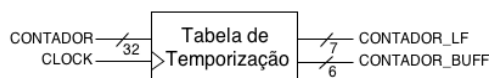


Figura 4.9: Bloco tabela de temporização

O valor do sinal CONTADOR, gerado pelo módulo de detecção de cores, é proporcional ao tempo necessário para que as notas de uma música se desloquem a partir de um ponto da tela a outro. Esta informação permite ao robô identificar a velocidade das notas.

No jogo GH3 são encontrados 16 níveis diferentes de velocidades para as notas de uma música. Para cada nível foi definido um valor mínimo e um valor máximo que determinam um intervalo de valores para os quais o valor de CONTADOR pode

se enquadrar; o intervalo cujo sinal de entrada CONTADOR se enquadra indica o nível de velocidade da música.

Internamente o bloco é composto por dois registradores cujos valores dos sinais de saída CONTADOR_LF (entrada para o bloco limitador de frequência) e CONTADOR_BUFF (entrada para o bloco *buffer*) são associados. Para cada nível de velocidade foi realizado um experimento manual de melhor ajuste utilizado para definir o valor que cada registrador deverá assumir de acordo com o intervalo em que o sinal de entrada CONTADOR se enquadrar.

4.2.3 Configuração e Monitoramento

O chip FPGA da placa Altera[®] DE2-35 representa o “cérebro” do robô. Ele é responsável por controlar todos os dispositivos de entrada e saída disponíveis na placa.

1. Chaves:

São utilizadas para a configuração manual do robô, para resetar o módulo DE2 TV, para resetar o contador gerado pelo módulo detecção de cores de identificação temporal e para atividades de debug.

2. *Displays* de 7 segmentos:

É utilizado para visualizar o valor de registradores de estado do robô e apresentação de informações de debug.

3. LEDS:

Os LEDs permitem monitorar visualmente os comandos enviados ao sistema de interação do robô e representa mais uma alternativa de debug de sinais digitais.

4. Memória (SRAM):

Parte da imagem da tela pode ser enviada a um bloco de memória. Seu conteúdo pode ser exportado para realização de análises de cores.

4.2.4 Sistema de Acionamento

O Sistema de acionamento recebe sinais elétricos de comando através de um conector GPIO e os transforma em estímulos mecânicos.

1. Componentes:

- Conector GPIO (*General Purpose Input Output*):

O conector de expansão GPIO é utilizado para transmissão de sinais de acionamento. O sinal de saída de cada pino do conector gera uma tensão de 3.3 V quando está em 1 e 0 V quando está em 0.

- Fonte de alimentação 700W, 12V, 39A; mostrada na Figura 4.10.



Figura 4.10: Fonte de alimentação

- Circuito de amplificação de sinal contruído em *protoboard*:

O circuito de amplificação, mostrado na Figura 4.11, é responsável pelo pressionamento dos botões do controle através do acionamento de solenoides; é formado por 5 subcircuitos iguais (Figura 4.12), cada um é composto por 1 resistor de $1\text{ k}\Omega$, 2 resistores de $100\ \Omega$, 1 diodo, 1 transistor BD-135, um optoacoplador e um solenoide.

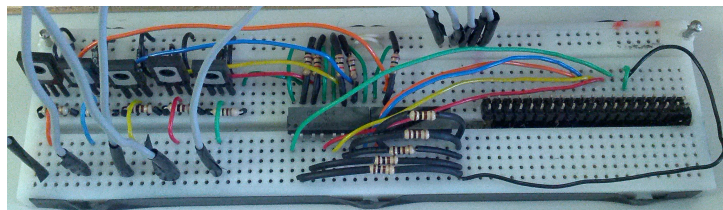


Figura 4.11: Subcircuito de acionamento

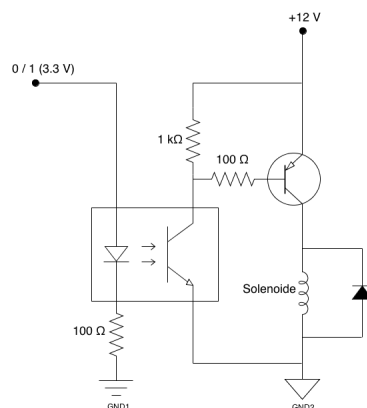


Figura 4.12: Esquemático do subcircuito de amplificação

- Atuadores (Solenoides):

Solenóide, mostrado na Figura 4.13 é o nome dado ao dispositivo eletromecânico composto por um fio condutor, enrolado em forma de bobina, que quando percorrido por corrente elétrica gera um campo magnético. No seu interior existe um “braço” metálico se desloca em decorrência da força aplicada pelo campo magnético gerado; e uma mola, que empurra o braço de volta à posição inicial quando a corrente elétrica para de circular pelo fio.

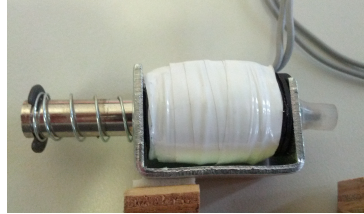


Figura 4.13: Solenoide

4.3 Grybot

Grybot é o nome dado ao robô apresentado neste projeto. A escolha do nome foi inspirada nas iniciais das palavras *Green, Red, Yellow, Blue, Orange*; significam: Verde, Vermelho, Amarelo, Azul e Laranja; nesta ordem elas se referem às cinco cores presentes no jogo GH3 que foram utilizadas para representar as notas.

Conforme apresentado na Figura 4.14, o sistema implementado é formado por componentes do jogo (2. Videogame, 9. Controle), do robô (3. Placa FPGA, 5. Circuito de amplificação, 6. Fonte de alimentação, 7. Suporte de madeira, 8. Solenóides) e componentes auxiliares (1. Notebook, 4. Monitor VGA).

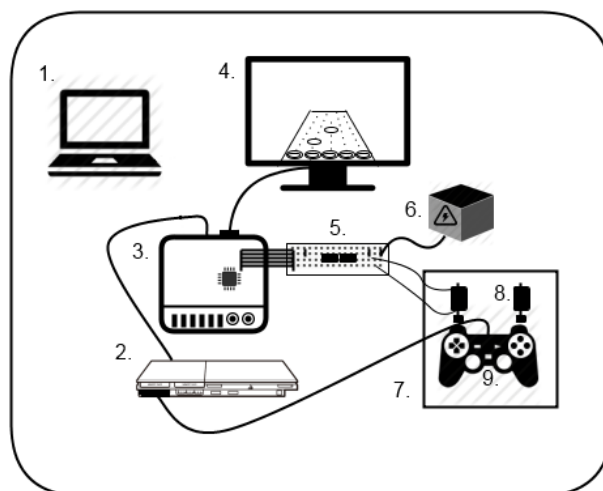


Figura 4.14: Esquemático do Grybot

O videogame PS2 é onde o GH3 é executado, a imagem e o áudio do jogo são enviados a partir do console através de cabos RCA; o vídeo é transmitido à placa FPGA (Seção 3.1) no formato descrito na Seção 3.2, e o módulo DE2_TV (Seção 3.4) é responsável por converter o padrão de entrada para o padrão VGA, aceito pelo monitor. O módulo detecção de cores (Detecção de Nota, Detecção de Rastro, Estimação de Velocidade), apresentado na Subseção 4.2.1, é responsável por detectar as notas analisando as componentes RGB dos pixels monitorados; o módulo Temporização (Limitador, Temporizador, *Buffer*) descrito na Subseção 4.2.2 define tempos de espera e frequência de acionamento dos solenoides de acordo com os parâmetros recebidos do módulo detecção de cores; o módulo configuração e monitoramento, detalhado na Subseção 4.2.4, é encarregado de captar a configuração definida para o robô através de chaves da placa DE2-35 e enviar feedback ao desenvolvedor por meio dos dispositivos periféricos (ex: LEDs e Displays de 7 segmentos). O conector GPIO, a fonte de alimentação, o circuito de amplificação, o suporte de madeira e os solenoides constituem o sistema de acionamento; a sua função é sustentar o posicionamento correto dos solenoides, captar o sinal de acionamento enviado pelo barramento GPIO e gerar corrente suficiente para os solenoides pressionarem os botões do controle.

O notebook não faz parte do sistema do robô pois é utilizado apenas para sintetizar o projeto e carregar o arquivo binário gerado com base no *netlist* (Seção 3.1); estando no chip FPGA, o robô opera sem que haja necessidade de um computador. O monitor é usado apenas para a apresentação da imagem do jogo na tela.

A Figura 4.15 mostra o diagrama de blocos do Grybot, detalhando o funcionamento do robô.

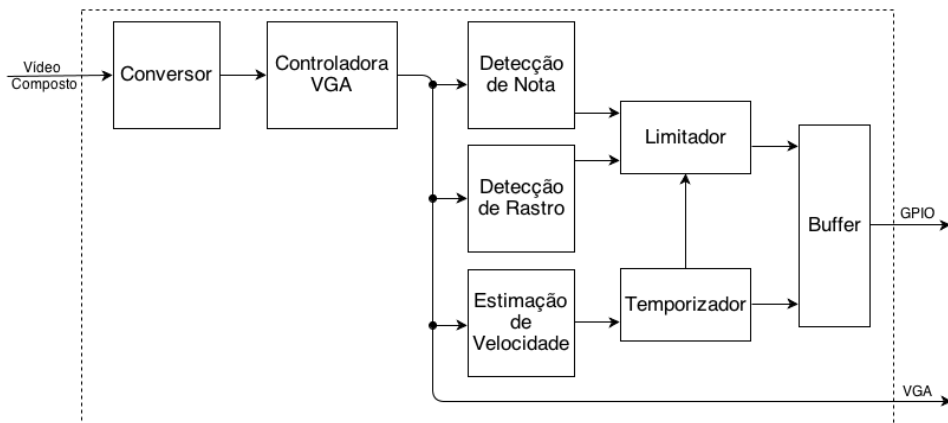


Figura 4.15: Diagrama de blocos do Grybot

A Figura 4.16 representa visualmente a utilização das unidades da FPGA. As tonalidades de azuis representam a densidade de utilização das unidades, quanto mais claras, menor utilização e quanto mais escuras, maior.

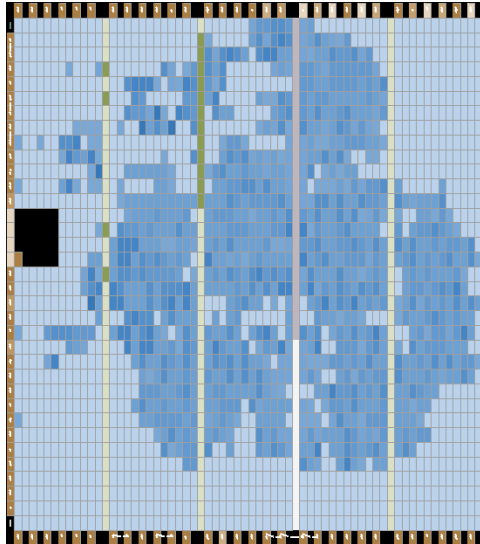


Figura 4.16: *FPGA Chip Planner*

A Figura 4.17 apresenta em números a taxa de uso dos elementos da FPGA.

Flow Status	Successful - Mon Sep 30 23:41:26 2013
Quartus II 64-Bit Version	12.1 Build 243 01/31/2013 SP 1 SJ Web Edition
Revision Name	DE2_TV
Top-level Entity Name	DE2_TV
Family	Cyclone II
Device	EP2C35F672C6
Timing Models	Final
Total logic elements	14,283 / 33,216 (43 %)
Total combinational functions	13,946 / 33,216 (42 %)
Dedicated logic registers	4,082 / 33,216 (12 %)
Total registers	4082
Total pins	426 / 475 (90 %)
Total virtual pins	0
Total memory bits	53,184 / 483,840 (11 %)
Embedded Multiplier 9-bit elements	44 / 70 (63 %)
Total PLLs	1 / 4 (25 %)

Figura 4.17: *FPGA Flow Summary*

Capítulo 5

Resultados

Este capítulo tem como foco principal expor as análises realizadas com o protótipo, listar todos os parâmetros de ajustes que foram definidos na implementação do robô, apresentar os resultados alcançados e comparar o desempenho obtido pelo sistema construído neste projeto, com o desempenho obtido por outros sistemas e por seres humanos.

5.1 Análise do Sistema Mecânico

Para garantir que o sistema mecânico de acionamento exerça corretamente sua função é necessário ter força suficiente para realizar o pressionamento dos botões do controle e responder com velocidade suficiente para acertar as notas consecutivas que atravessavam rapidamente o braço da guitarra.

A fim de verificar esses parâmetros foram realizadas medidas da quantidade máxima de notas de mesma cor que atravessavam o braço da guitarra por segundo.

Utilizando um adaptador para o controle do videogame que permite conectá-lo ao computador pela entrada USB, foi escrito um programa em linguagem C++, baseado na biblioteca SDL, que captura o evento gerado pelo pressionamento dos botões do controle e realiza a contagem do número de vezes que cada botão é pressionado por segundo.

Um circuito digital descrito em linguagem Verilog foi construído e carregado na placa DE2-35. Este circuito envia comandos de frequência crescente ao sistema de pressionamento fazendo os solenoides operarem com um ciclo de trabalho de 81,25%, isto é, 81,25% do tempo de 1 ciclo o solenoide é mantido acionado e em 18,75% é mantido em repouso. Esse valor foi escolhido pois representa o mesmo ciclo adotado para o bloco limitador de frequência.

Cada solenoide ficou encarregado pelo pressionamento de um botão do controle. A captura de notas com cores diferentes é relativamente simples, ainda que muito próximas umas das outras. O problema mais desafiador é o pressionamento de notas de uma mesma cor quando aparecem sequencialmente e próximas umas das outras, exigindo alta frequência de acionamento de um solenoide.

O método adotado para medir a taxa máxima de notas com mesma cor consecutivas por segundo foi baseado na análise de modelos adaptados de partituras para músicas do jogo (Figura 5.1), disponibilizadas no forum do site ScoreHero [32].

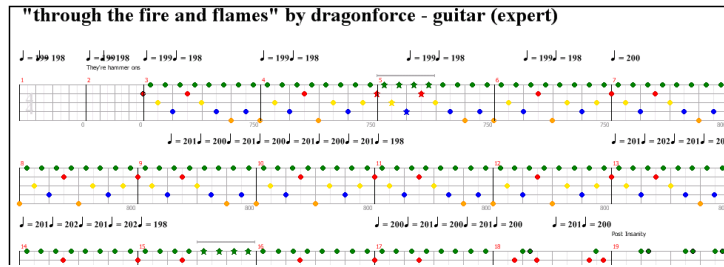


Figura 5.1: Partitura da música *Through The Fire And Flames*

A maior taxa observada para a música mais desafiadora do jogo, *Through The Fire And Flames* da banda Dragon Force, foi de 16 notas de mesma cor por segundo. Todos os solenoides foram capazes de atingir individualmente a taxa de 18 pressionamentos por segundo. Assim, é possível concluir que o sistema mecânico de acionamento construído pode ser utilizado para o propósito escolhido.

Durante a fase de testes do robô foi verificado que o posicionamento dos solenoides e o número de solenoides acionados simultaneamente influenciam na força resultante, afetando também a taxa de frequência máxima de pressionamento dos botões.

Foi observado que a distância na qual a extremidade do braço do solenoide se encontra em relação aos botões do controle é decisivo para o correto funcionamento do sistema (Figura 5.2). Quando o solenóide é posicionado de 2 a 3 milímetros distante da posição ideal, se tornam mais fracos ou até mesmo incapazes de acionar os botões do controle.



Figura 5.2: Controle DualShock2 e solenoides

Posicionar os solenoides na direção vertical e apontados para baixo (Figura 5.2), permite que o robô obtenha desempenho superior ao posicionamento horizontal. Um dos motivos que explica este fenômeno é a força lateral aplicada ao solenoide gerada pela gravidade, que faz com que o atrito existente no braço do dispositivo seja maior e atrapalhe sua movimentação.

O sistema de pressionamento é alimentado por uma fonte de tensão não ideal, isto é, a sua tensão de saída varia de acordo com a corrente requerida pelo circuito. Considerando que cada solenoide requer uma quantidade razoável de corrente para seu funcionamento,

conclui-se que quanto maior é o número de solenoides acionados simultaneamente, menor fica a tensão da fonte, por consequência, menor se torna a corrente fornecida para cada solenoide. Dado que a força magnética é proporcional à intensidade de corrente que circula na bobina do solenoide, a força exercida pelo braço sobre os botões do controle também sofre redução.

Os dados apresentados na Tabela 5.1 mostram a variação das medidas de tensão de saída da fonte extraídas quando 0, 1, 2, 3, 4 e 5 solenoides são acionados simultaneamente.

Tabela 5.1: Número de solenoides vs Tensão

Nº Solenoides	0	1	2	3	4	5
Tensão (V)	12,10	11,71	11,36	11,03	10,73	10,45

A Tabela 5.2 apresenta as medidas de corrente, em Ampère, que circulam em cada solenoide de acordo com o número de solenoides acionados simultaneamente.

Tabela 5.2: Quantidade de solenoides vs Corrente

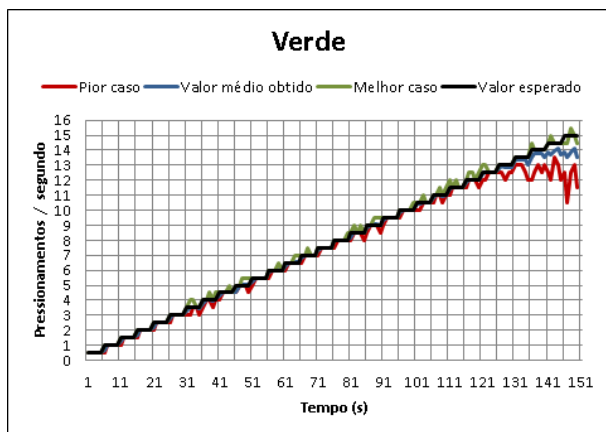
Qtd. Sol.	Verde	Vermelho	Amarelo	Azul	Laranja
1	0,88	0,91	0,90	0,88	0,93
2	0,85	0,91	0,86	0,86	0,90
3	0,82	0,88	0,84	0,83	0,87
4	0,80	0,85	0,82	0,81	0,84
5	0,78	0,83	0,79	0,79	0,81

Os gráficos das figuras 5.3a, 5.3b, 5.4a, 5.4b, 5.5a, 5.5b, 5.6a, 5.6b, 5.7a e 5.7b foram construídos, a partir de 10 amostras do teste de taxa de frequência a um ciclo de trabalho de acionamento de 50%. A métrica “Pior caso” representa o valor mínimo obtido dentre as amostras; a métrica “Valor médio obtido” corresponde ao cálculo da média dos valores; a métrica “Melhor caso” se refere ao valor máximo. A métrica “Valor esperado” contém a taxa de pressionamentos que deveria ser captada pelo controle se o sistema fosse capaz de pressionar os botões conforme o sinal de acionamento enviado.

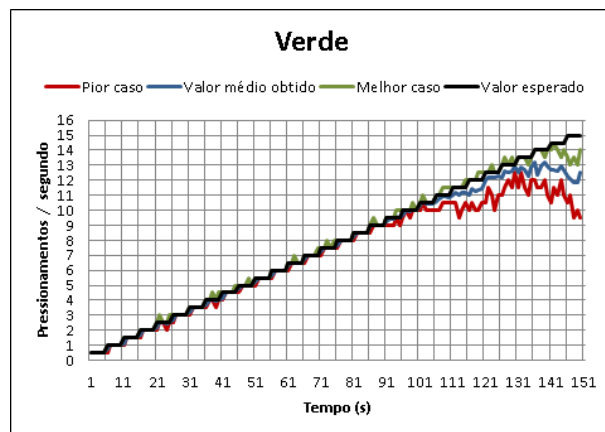
Os gráficos apresentam a resposta dos botões ao estímulo do acionamento dos solenoides a partir de taxas de frequência crescentes de 0,5 pressionamentos por segundo a cada cinco amostras. As figuras dispostas à esquerda se referem a medidas obtidas do acionamento de apenas um solenoide, enquanto os gráficos à direita foram extraídos durante o acionamento simultâneo dos cinco solenoides do sistema.

Analisando os resultados apresentados é possível notar que o valor médio dos gráficos posicionados à esquerda se mantiveram mais próximos do valor esperado por mais tempo que os gráficos à direita. Este efeito ocorre devido a diminuição da corrente que circula nos solenoides e consequente redução da força de pressionamento.

Os dados apresentados nos gráficos das Figuras 5.7a e 5.7b se mostram bem diferentes dos demais. Isto ocorre pois as medidas extraídas para compor esses gráficos foram obtidas do único solenoide que precisa ser fixado perpendicularmente aos demais (Figura 5.2), na posição horizontal (correspondente à nota laranja), o que demonstra grande perda de desempenho frente aos outros solenoides posicionados verticalmente.

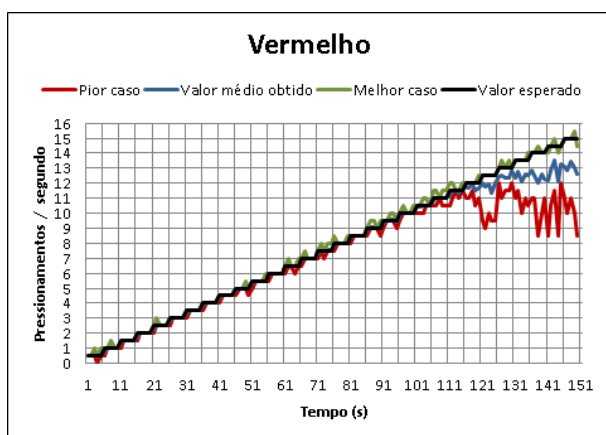


(a)

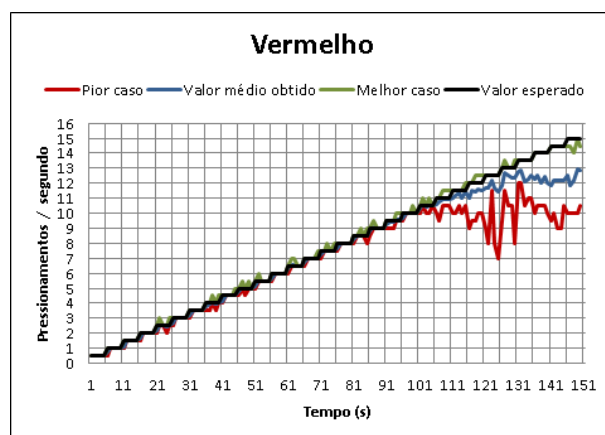


(b)

Figura 5.3: Solenoide correspondente à nota verde. (a) 1 solenoide (b) 5 solenoides

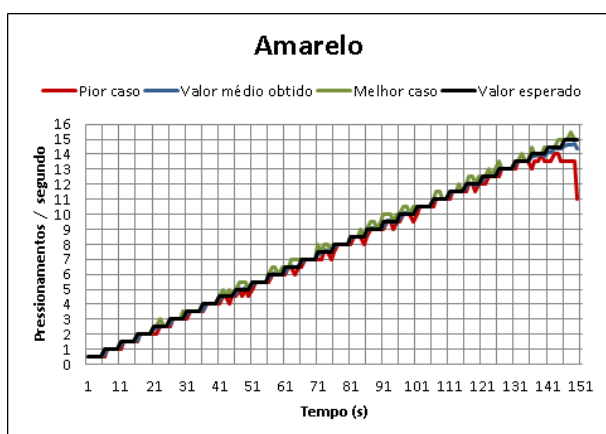


(a)

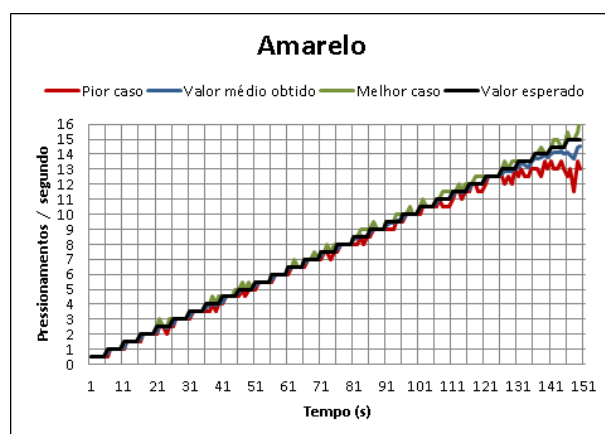


(b)

Figura 5.4: Solenoide correspondente à nota vermelha. (a) 1 solenoide (b) 5 solenoides

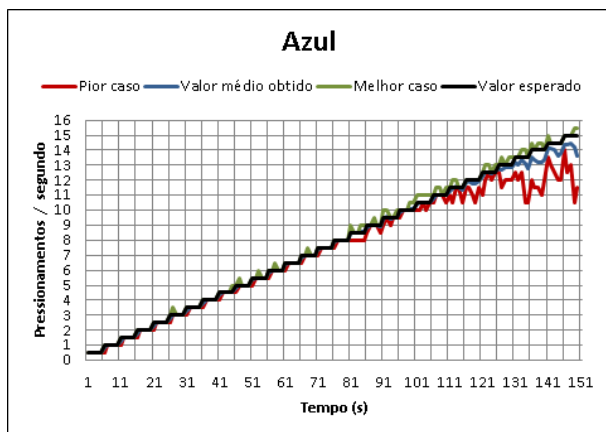


(a)

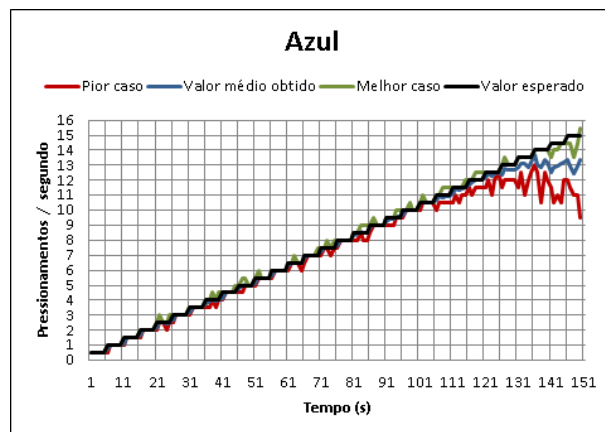


(b)

Figura 5.5: Solenoide correspondente à nota amarela. (a) 1 solenoide (b) 5 solenoides

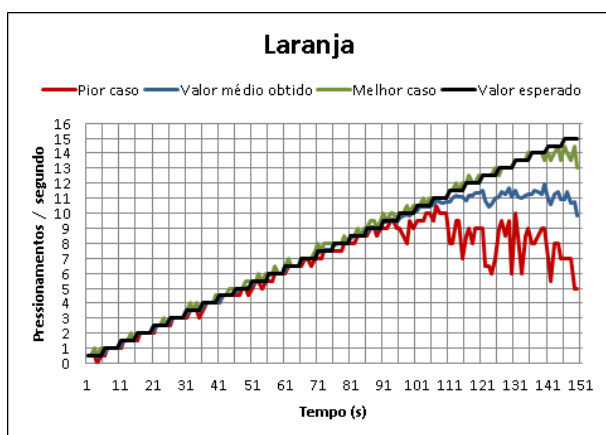


(a)

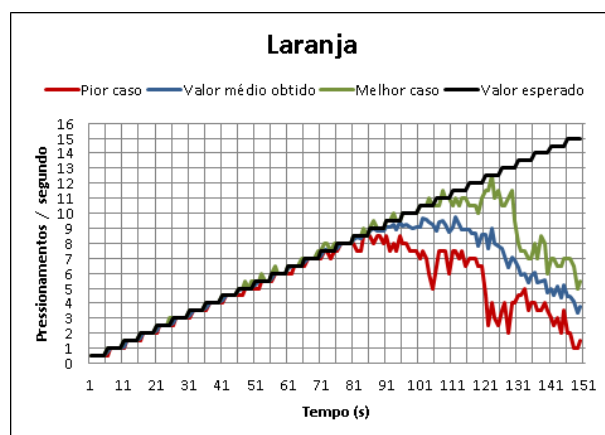


(b)

Figura 5.6: Solenoide correspondente à nota azul. (a) 1 solenoide (b) 5 solenoides



(a)



(b)

Figura 5.7: Solenoide correspondente à nota laranja. (a) 1 solenoide (b) 5 solenoides

Fatores como atrito, temperatura e umidade do ar exercem consideráveis influências sobre qualquer sistema mecânico, e por conseguinte no desempenho dos solenoides, entretanto os impactos gerados por estes fatores não foram medidos neste projeto.

5.2 Definição dos Parâmetros

No Capítulo 4 foram definidos vários parâmetros utilizados nos métodos de captura da imagem, detecção de notas e cálculo de temporização utilizado pelo robô para gerar os sinais de acionamento dos solenoides. Nesta seção são apresentados os valores assumidos pelos parâmetros definidos na construção do protótipo.

O método de captura da imagem descrito na Seção 4.2.1 se baseia no monitoramento dos valores RGBs de pixels posicionados em locais específicos da tela do jogo.

Posicionamento dos pixels monitorados pelo módulo detecção de cores:

- Grupo de pixels para detecção de notas em geral.

A Tabela 5.3 apresenta em coordenadas (linha,coluna) a posição dos pixels utilizados para a detecção de notas de cada cor do grupo mostrado na Figura 4.1.

Tabela 5.3: Posições dos pixels do grupo de detecção de notas em geral

Verde	Vermelho	Amarelo	Azul	Laranja
(245,330)	(285,330)	(325,330)	(365,330)	(405,330)

- Grupo de pixels para detecção de rastros (“notas contínuas”).

A Tabela 5.4 apresenta a posição dos pixels utilizados para a detecção de rastros. A primeira coluna indica o número do pixel em relação a um conjunto de 13 pixels monitorados para cada cor, conforme mostrado na Figura 4.2.

Tabela 5.4: Posições dos pixels para detecção de rastros (“notas contínuas”)

Pixel	Verde	Vermelho	Amarelo	Azul	Laranja
1	(250,318)	(287,318)	(325,318)	(364,318)	(401,318)
2	(250,317)	(287,317)	(325,317)	(363,317)	(401,317)
3	(251,316)	(287,316)	(325,316)	(363,316)	(400,316)
4	(251,315)	(288,315)	(325,315)	(363,315)	(400,315)
5	(251,314)	(288,314)	(325,314)	(363,314)	(400,314)
6	(252,313)	(288,313)	(325,313)	(363,313)	(399,313)
7	(252,312)	(288,312)	(325,312)	(362,312)	(399,312)
8	(252,311)	(288,311)	(325,311)	(362,311)	(399,311)
9	(253,310)	(289,310)	(325,310)	(362,310)	(398,310)
10	(253,309)	(289,309)	(325,309)	(362,309)	(398,309)
11	(253,308)	(289,308)	(325,308)	(362,308)	(398,308)
12	(254,307)	(289,307)	(325,307)	(361,307)	(397,307)
13	(254,306)	(289,306)	(325,306)	(361,306)	(397,306)

- Grupo de pixels utilizados para medida da temporização.

A Tabela 5.5 apresenta a posição dos pixels utilizados para medida da temporização. A primeira coluna indica o número do pixel em relação a um conjunto de 2 pixels monitorados para cada cor, conforme mostrado na Figura 4.3.

Tabela 5.5: Posições dos pixels utilizados para medida da temporização

Pixel	Verde	Vermelho	Amarelo	Azul	Laranja
1	(245,330)	(285,330)	(325,330)	(365,330)	(405,330)
2	(212,411)	(269,411)	(325,411)	(382,411)	(438,411)

- Intervalos de valores RGB para detecção de notas pela cor.

A Tabela 5.6 apresenta os intervalos de valores RGB utilizados para identificar notas através da cor. As linhas Verde, Vermelho, Amarelo, Azul e Laranja representam as regiões da guitarra por onde as notas das respectivas cores atravessam o braço da guitarra. Os valores da tabela são apresentados no formato ($\langle \text{Red} \rangle, \langle \text{Green} \rangle, \langle \text{Blue} \rangle$), que representa o valor das componentes RGB. A segunda coluna da tabela indica a limite inferior do intervalo de cores, a terceira indica o limite superior.

Tabela 5.6: Intervalos de valores RGB para detecção de notas pela cor

Nota	C. Mínima	C. Máxima
Verde	(0,700,0)	(1023,1023,1023)
Vermelho	(700,0,0)	(1023,1023,1023)
Amarelo	(700,600,0)	(1023,1023,1023)
Azul	(0,500,700)	(1023,1023,1023)
Laranja	(700,380,0)	(1023,1023,1023)

Além dos intervalos definidos na Tabela 5.6, a identificação de notas também ocorre quando as componentes RGB do pixel monitorado se enquadra no intervalo (570,570,570), (1023,1023,1023). Esta condição foi adotada para definir detecção de notas pelo fato de que são os únicos objetos do braço da guitarra (com exceção de linhas brancas) que possuem intensidade alta de brilho.

- Esfera de detecção do rastro das notas pela cor.

A Tabela 5.7 apresenta os valores correspondentes aos pontos centrais das esferas escolhidas para determinar as faixas de cores que indicam a detecção do rastro de notas contínuas para cada cor. Para todas as cores definiu-se experimentalmente um limiar para a distância da cor de $\sqrt{30000}$.

Tabela 5.7: Cores centrais definidas para esferas de detecção de cor

Nota	C. Principal
Verde	(20,1023,20)
Vermelho	(1023,390,160)
Amarelo	(1023,920,0)
Azul	(100,870,1023)
Laranja	(1023,592,40)

- Intervalos de valores RGB para detecção de notas pela cor utilizados para medida da temporização.

A Tabela 5.8 apresenta os intervalos de valores RGB utilizados para identificar notas através da cor e definir o momento de início e paralização da contagem que calcula os parâmetros de temporização. A segunda coluna da tabela indica o limite inferior do intervalo, a terceira indica o limite superior.

Tabela 5.8: Intervalos de valores RGB para detecção de notas pela cor para medidas de temporização

Nota	C. Mínima	C. Máxima
Verde	(0,704,0)	(1023,1023,1023)
Vermelho	(768,0,0)	(1023,1023,1023)
Amarelo	(680,612,0)	(1023,1023,512)
Azul	(0,448,768)	(1023,1023,1023)
Laranja	(768,320,0)	(1023,1023,1023)

- Parâmetros de medida da velocidade da música.

O tempo que uma nota leva para sair da posição do pixel 1 descrito na Tabela 5.5 e chegar até o pixel 2 é utilizado para descobrir dinamicamente a velocidade da música tocada. A Tabela 5.9 faz associações entre os níveis de dificuldade do jogo e velocidade das músicas. As velocidades são associadas às músicas através dos intervalos de tempo apresentados na Tabela 5.10.

Tabela 5.9: Relação entre velocidades e níveis de dificuldades

Velocidade	Nível - Velocidade
1	Expert - Full Speed
2	Hard - Full Speed
3	Expert - Slow
4	Medium - Full Speed
5	Hard - Slow
6	Expert - Slower
7	Easy - Full Speed
8	Medium - Slow
9	Hard - Slower
10	Expert - Slowest
11	Medium - Slower
12	Easy - Slow
13	Hard - Slowest
14	Easy - Slower
15	Medium - Slowest
16	Easy - Slowest

Tabela 5.10: Relação entre intervalo de tempo e velocidade

Velocidade	Mínimo	Máximo
1	0,1851852	0,2801571
2	0,2801572	0,3117192
3	0,3117193	0,3341729
4	0,334173	0,3592001
5	0,3592002	0,3787988
6	0,3787989	0,4051025
7	0,4051026	0,429633
8	0,4296331	0,4422581
9	0,4422582	0,4815252
10	0,4815253	0,5160582
11	0,5160583	0,5254475
12	0,5254476	0,5630872
13	0,5630873	0,6150193
14	0,6150194	0,6569991
15	0,6569992	0,7614056
16	0,7614057	1,037037

- Parâmetros de temporização do buffer.

A detecção de notas da música antecede o momento em que o botão do controle deve ser pressionado. O tempo de espera em segundos entre a identificação de uma nota e o envio do comando de acionamento dos solenoides é determinado pelo buffer. A Tabela 5.11 apresenta, em segundos, o tempo de espera definido experimentalmente para cada velocidade existente no jogo.

Tabela 5.11: Tempos de espera do *Buffer*

Velocidade	Tempo de Espera
1	0,133
2	0,150
3	0,217
4	0,217
5	0,267
6	0,250
7	0,317
8	0,350
9	0,350
10	0,400
11	0,433
12	0,433
13	0,533
14	0,550
15	0,617
16	0,800

- Parâmetros de temporização do limitador de frequência.

O limitador de frequência foi criado para que o robô acertasse sequências de notas de mesma cor e muito próximas. A Tabela 5.12 registra os tempos definidos no limitador para Verificação (coluna 2), Pressionamento (coluna 3), Repouso (coluna 4) e Correção de Perspectiva (coluna 5) para cada velocidade (coluna 1) de música existente no jogo.

Tabela 5.12: Tempos do limitador de frequência

Velocidade	T. Verificação	T. Pressionamento	T. Repouso	T. Perspectiva
1	0,00128	0,06656	0,01536	0,128
2	0,00152	0,07904	0,01824	0,152
3	0,00148	0,07696	0,01776	0,148
4	0,00208	0,10816	0,02496	0,208
5	0,00172	0,08944	0,02064	0,172
6	0,0018	0,0936	0,0216	0,18
7	0,0024	0,1248	0,0288	0,24
8	0,00232	0,12064	0,02784	0,232
9	0,00196	0,10192	0,02352	0,196
10	0,00236	0,12272	0,02832	0,236
11	0,00256	0,13312	0,03072	0,256
12	0,00256	0,13312	0,03072	0,256
13	0,0026	0,1352	0,0312	0,26
14	0,00284	0,14768	0,03408	0,284
15	0,00312	0,16224	0,03744	0,312
16	0,00384	0,19968	0,04608	0,384

5.3 Desempenho Obtido

Ao final de cada música é disponibilizado na tela o total de pontos atingidos (*Score*), o número máximo de notas consecutivas acertadas (*Streak*) e a porcentagem do total de notas acertadas em relação ao total de notas da música (*Hits*) alcançado pelo jogador.

- Tabelas de acertos das músicas por nível de dificuldade.

O robô foi designado a tocar, no modo *Quick Play*, cada música do jogo GH3. Os resultados obtidos estão listados nas Tabelas 5.13, 5.14, 5.15 e 5.16 a seguir:

Tabela 5.13: Desempenho obtido no nível *Easy*

Nome da Música	Score	Streak	Hits
Slow Ride	62.006	123	100%
Talk Dirty to Me	78.501	200	100%
Hit Me With Your Best Shot	39.149	79	100%
Story Of My Life	98.144	63	98%
Rock and Roll All Nite	46.901	68	98%
Sabotage	46.336	163	99%
Mississippi Queen	23.017	54	99%
School's Out	51.755	92	100%
Sunshine of Your Love	81.298	246	99%
Barracuda	79.829	91	100%
Bulls on Parade	68.173	92	100%
Reptilia	76.144	156	99%
When You Were Young	62.146	104	100%
Miss Murder	51.870	181	100%
The Seeker	53.729	71	100%
Lay Down	60.521	138	100%
Paint It Black	71.445	150	99%
Suck My Kiss	35.185	59	99%
Paranoid	39.540	47	100%
Anarchy ub U.K.	74.441	127	100%
Kool Thing	41.473	31	100%
My Name Is Jonas	73.327	150	99%
Even Flow	67.196	101	99%
Cities On Flame with Rock & Roll	69.733	208	100%
Holiday In Cambodia	70.104	75	100%
Rock You Like A Hurr cane	76.601	156	100%
Same Old Song and Dance	55.536	84	100%
La Grange	86.232	89	100%
Welcome To The Jungle	78.709	112	100%
Helicopter	56.129	70	99%
Black Magic Woman	76.418	88	99%
Cherub Rock	100.286	299	100%
Black Sunshine	74.944	88	99%

The Metal	52.125	119	99%
Pride and Joy	81.393	107	99%
Monsters	67.666	101	99%
Before I Forget	88.498	104	100%
Stricken	83.350	120	99%
3's & 7's	80.156	123	99%
Knights of Cydonia	116.507	183	100%
Cult Of Personality	91.768	150	100%
Raining Blood	113.785	265	100%
Cliffs Of Dover	67.354	131	99%
The Number of the Beast	127.070	124	99%
One	117.677	122	100%
Avalancha	130.211	189	100%
In The Belly Of A Shark	44.191	53	96%
Can't Be Saved	54.229	165	100%
Closer	54.517	125	100%
Don't Hold Back	63.994	59	100%
Down 'N Dirty	111.175	250	100%
F.C.P.R.E.M.I.X	75.469	89	99%
Generation Rock	32.368	67	99%
Go That Far	40.143	128	100%
Hier Kommt Alex	88.103	133	100%
I'm In The Band	52.887	59	99%
Impulse	76.033	116	100%
In Love	62.819	71	98%
Mauvais Garçon	44.224	50	100%
Metal Heavy Lady	29.910	77	100%
Minus Celsius	52.971	44	98%
My Curse	92.944	122	99%
Nothing For Me Here	50.478	93	98%
Prayer Of The Refugee	58.633	112	100%
Radio Song	51.119	75	100%
Ruby	47.719	102	100%
She Bangs The Drums	55.110	98	100%
Take This Life	49.694	63	97%
The Way It Ends	94.519	129	99%
Through The Fire And Flames	178.226	179	99%

Tabela 5.14: Desempenho obtido no nível *Medium*

Nome da Música	Score	Streak	Hits
Slow Ride	93.456	97	99%
Talk Dirty to Me	93.005	253	99%
Hit Me With Your Best Shot	53.703	67	98%

Story Of My Life	122.608	79	97%
Rock and Roll All Nite	67.369	83	98%
Sabotage	96.689	195	99%
Mississippi Queen	54.721	92	99%
School's Out	75.181	194	100%
Sunshine of Your Love	112.784	100	99%
Barracuda	146.227	287	100%
Bulls on Parade	107.713	123	99%
Reptilia	116.850	267	99%
When You Were Young	109.455	122	99%
Miss Murder	91.894	287	99%
The Seeker	69.584	64	99%
Lay Down	134.144	518	99%
Paint It Black	120.353	261	99%
Suck My Kiss	90.930	146	98%
Paranoid	62.907	56	98%
Anarchy ub U.K.	132.819	172	99%
Kool Thing	135.072	122	99%
My Name Is Jonas	128.981	183	99%
Even Flow	99.533	114	99%
Cities On Flame with Rock & Roll	80.061	95	99%
Holiday In Cambodia	134.108	190	99%
Rock You Like A Hurr cane	118.778	189	99%
Same Old Song and Dance	85.025	132	99%
La Grange	115.568	144	99%
Welcome To The Jungle	106.626	126	99%
Helicopter	117.771	136	99%
Black Magic Woman	103.276	94	99%
Cherub Rock	163.776	140	98%
Black Sunshine	96.021	163	98%
The Metal	80.943	166	99%
Pride and Joy	124.543	128	99%
Monsters	131.312	190	99%
Before I Forget	153.830	199	99%
Stricken	181.691	186	99%
3's & 7's	150.402	242	99%
Knights of Cydonia	196.346	235	99%
Cult Of Personality	130.594	146	100%
Raining Blood	154.789	239	99%
Cliffs Of Dover	107.218	132	99%
The Number of the Beast	159.202	185	98%
One	264.312	241	98%
Avalancha	169.242	109	99%
In The Belly Of A Shark	79.516	187	98%

Can't Be Saved	81.865	134	99%
Closer	101.628	107	99%
Don't Hold Back	130.317	180	98%
Down 'N Dirty	136.300	225	99%
F.C.P.R.E.M.I.X	171.701	243	99%
Generation Rock	83.997	139	98%
Go That Far	92.601	248	99%
Hier Kommt Alex	168.850	323	99%
I'm In The Band	92.589	83	98%
Impulse	120.025	306	100%
In Love	111.480	293	99%
Mauvais Garçon	100.551	86	98%
Metal Heavy Lady	69.962	119	99%
Minus Celsius	90.599	77	99%
My Curse	144.603	282	99%
Nothing For Me Here	103.330	117	99%
Prayer Of The Refugee	125.148	176	99%
Radio Song	76.341	102	98%
Ruby	63.625	73	99%
She Bangs The Drums	81.112	54	98%
Take This Life	153.604	244	98%
The Way It Ends	162.697	162	99%
Through The Fire And Flames	345.769	266	99%

Tabela 5.15: Desempenho obtido no nível *Hard*

Nome da Música	Score	Streak	Hits
Slow Ride	91.186	106	99%
Talk Dirty to Me	148.439	112	98%
Hit Me With Your Best Shot	87.792	75	96%
Story Of My Life	233.768	104	98%
Rock and Roll All Nite	78.218	74	98%
Sabotage	176.369	183	99%
Mississippi Queen	57.900	51	98%
School's Out	93.378	125	99%
Sunshine of Your Love	128.671	83	98%
Barracuda	184.669	205	99%
Bulls on Parade	159.707	159	99%
Reptilia	163.917	186	99%
When You Were Young	155.055	120	99%
Miss Murder	156.116	100	97%
The Seeker	140.328	110	98%
Lay Down	149.515	125	99%
Paint It Black	154.251	97	99%

Suck My Kiss	140.828	171	98%
Paranoid	193.070	132	99%
Anarchy ub U.K.	165.075	96	98%
Kool Thing	190.053	237	99%
My Name Is Jonas	176.270	119	98%
Even Flow	143.194	128	99%
Cities On Flame with Rock & Roll	117.992	114	99%
Holiday In Cambodia	284.142	252	99%
Rock You Like A Hurrcafe	201.904	268	99%
Same Old Song and Dance	111.044	122	99%
La Grange	148.637	153	99%
Welcome To The Jungle	136.965	109	98%
Helicopter	153.870	188	98%
Black Magic Woman	139.867	148	99%
Cherub Rock	211.633	120	97%
Black Sunshine	135.056	213	100%
The Metal	131.380	169	100%
Pride and Joy	193.001	152	98%
Monsters	233.094	236	99%
Before I Forget	228.939	109	98%
Stricken	266.768	190	99%
3's & 7's	223.350	168	98%
Knights of Cydonia	235.148	337	95%
Cult Of Personality	180.518	186	99%
Raining Blood	221.384	275	99%
Cliffs Of Dover	161.884	153	99%
The Number of the Beast	309.264	290	99%
One	299.516	268	96%
Avalancha	225.968	108	99%
In The Belly Of A Shark	141.667	222	99%
Can't Be Saved	139.201	124	99%
Closer	101.820	87	98%
Don't Hold Back	153.846	109	98%
Down 'N Dirty	265.756	175	99%
F.C.P.R.E.M.I.X	227.540	178	98%
Generation Rock	139.526	143	98%
Go That Far	137.612	144	100%
Hier Kommt Alex	237.072	146	99%
I'm In The Band	142.883	118	98%
Impulse	164.112	253	99%
In Love	150.182	163	98%
Mauvais Garçon	156.534	74	97%
Metal Heavy Lady	92.547	167	99%
Minus Celsius	136.882	120	98%

My Curse	167.891	242	99%
Nothing For Me Here	137.845	111	98%
Prayer Of The Refugee	201.689	172	98%
Radio Song	96.428	101	99%
Ruby	83.808	97	99%
She Bangs The Drums	150.605	126	98%
Take This Life	303.838	172	99%
The Way It Ends	277.553	317	99%
Through The Fire And Flames	430.087	199	96%

Tabela 5.16: Desempenho obtido no nível *Expert*

Nome da Música	Score	Streak	Hits
Slow Ride	113.645	105	98%
Talk Dirty to Me	179.029	113	97%
Hit Me With Your Best Shot	89.396	82	97%
Story Of My Life	254.811	154	98%
Rock and Roll All Nite	113.370	93	98%
Sabotage	186.854	167	98%
Mississippi Queen	65.230	74	98%
School's Out	91.470	74	98%
Sunshine of Your Love	161.158	110	98%
Barracuda	195.812	189	99%
Bulls on Parade	174.992	160	98%
Reptilia	166.105	200	98%
When You Were Young	150.455	110	98%
Miss Murder	179.135	114	98%
The Seeker	170.056	168	98%
Lay Down	167.220	127	96%
Paint It Black	194.302	143	98%
Suck My Kiss	140.350	180	97%
Paranoid	195.544	125	98%
Anarchy ub U.K.	197.814	109	97%
Kool Thing	212.078	212	99%
My Name Is Jonas	212.220	315	98%
Even Flow	213.908	279	99%
Cities On Flame with Rock & Roll	166.003	194	99%
Holiday In Cambodia	324.065	349	98%
Rock You Like A Hurrcaene	221.065	250	98%
Same Old Song and Dance	140.379	104	98%
La Grange	168.077	182	99%
Welcome To The Jungle	203.698	169	98%
Helicopter	175.533	121	97%
Black Magic Woman	148.110	132	98%

Cherub Rock	261.689	145	96%
Black Sunshine	195.877	334	99%
The Metal	187.988	218	98%
Pride and Joy	214.823	95	97%
Monsters	251.610	296	97%
Before I Forget	268.524	85	96%
Stricken	260.014	151	96%
3's & 7's	208.254	95	96%
Knights of Cydonia	266.779	343	96%
Cult Of Personality	246.458	355	99%
Raining Blood	224.761	426	97%
Cliffs Of Dover	221.476	192	99%
The Number of the Beast	371.422	157	97%
One	432.545	245	98%
Avalancha	226.772	160	97%
In The Belly Of A Shark	190.254	137	98%
Can't Be Saved	140.477	150	97%
Closer	98.486	82	96%
Don't Hold Back	196.625	150	96%
Down 'N Dirty	307.599	268	98%
F.C.P.R.E.M.I.X	243.345	149	97%
Generation Rock	158.758	147	97%
Go That Far	158.742	230	99%
Hier Kommt Alex	235.672	179	97%
I'm In The Band	153.497	86	96%
Impulse	213.176	265	99%
In Love	180.940	312	98%
Mauvais Garçon	206.150	83	96%
Metal Heavy Lady	116.756	225	99%
Minus Celsius	132.906	82	96%
My Curse	256.214	235	98%
Nothing For Me Here	187.945	191	98%
Prayer Of The Refugee	171.215	73	96%
Radio Song	86.810	87	97%
Ruby	119.637	164	99%
She Bangs The Drums	171.239	96	97%
Take This Life	367.596	242	98%
The Way It Ends	347.799	175	97%
Through The Fire And Flames	669.898	512	98%

- Tabela de acertos da música *Through The Fire And Flames* (TTFAF).

A música TTFAF pode ser considerada a mais difícil do GH3; representa um grande desafio ao jogador, pois é longa, rápida e apresenta diversas combinações de notas.

Assumindo que também se trata do pior caso para o robô foi efetuada uma análise mais minuciosa.

Cinco amostras de resultados foram obtidas a partir da música TTFAF em cada nível de velocidade do modo treino. A Tabela 5.17 apresenta na coluna *Score*, a média da pontuação extraída de cada amostra. A coluna Desvio (%) indica o percentual que o desvio padrão amostral representa em relação à média.

Tabela 5.17: Desempenho na música *Through The Fire And Flames* modo *Training*

Nível - Velocidade	Score	Desvio (%)	Streak	Desvio (%)	Hits
Easy - Full Speed	213.281,0	0,11%	649,0	0%	100%
Easy - Slow	211.961,8	0,04%	649,0	0%	100%
Easy - Slower	211.780,8	0,23%	649,0	0%	100%
Easy - Slowest	212.936,6	0,19%	649,0	0%	99%
Medium - Full Speed	383.412,4	0,31%	641,0	0%	100%
Medium - Slow	384.619,4	0,21%	644,6	0,76%	100%
Medium - Slower	383.610,	0,24%	641,0	0%	100%
Medium - Slowest	381.027,2	0,69%	641,6	0,77%	100%
Hard - Full Speed	455.939,2	2,09%	621,4	25,15%	97%
Hard - Slow	523.695,6	0,72%	775,4	26,96%	98%
Hard - Slower	459.030,6	2,1%	359,2	9,91%	99%
Hard - Slowest	483.381,8	1,69%	434,4	13,66%	99%
Expert - Full Speed	698.622,6	1,69%	1.118,2	37,16%	98%
Expert - Slow	739.835,6	0,26%	662,4	28,48%	99%
Expert - Slower	708.992,6	1,14%	469,2	7,25%	99%
Expert - Slowest	746.784,6	0,9%	1.066,4	32,62%	99%

5.4 Comparações

Nesta seção são apresentados os resultados comparativos realizados com o protótipo construído neste projeto com outros robôs jogadores de GH3. Grybot é único robô que utiliza o controle convencional (DualShock2) do videogame (PS2) para jogar, todos os demais encontrados nesta pesquisa usam o controlador em formato de guitarra. Assim, os outros robôs não sofrem da redução de desempenho devido a ter um solenoide posicionado perpendicularmente aos demais.

5.4.1 Grybot vs Slashbot

Os dois robôs utilizam o sinal de vídeo analógico para realizar a análise dos pixels da imagem e enviar sinal de acionamento a um sistema mecânico composto de solenóides. Ao contrário do Slashbot, Grybot não possui um acionador destinado ao pressionamento do botão que dispara o *Star Power* (“Especial”).

A Tabela 5.18 mostra um comparativo de desempenho obtido pelos dois robôs no modo *Quick Play*, música *Cliffs of Dover*; para o comparativo realizado com a música *My*

Name is Jonas os dados do robô Slashbot foram extraídos do modo *Multiplayer*, enquanto para Grybot, por não possuir implementação para esse modo, foi utilizado *Quick Play*.

Considerando as informações apresentadas na Tabela 5.18 pode-se inferir que Grybot apresentou melhor performance em relação ao Slashbot.

Tabela 5.18: Grybot vs Slashbot [35, 36]

	Grybot			Slashbot		
	Score	Streak	Hits	Score	Streak	Hits
Cliffs Of Dover	221476	192	99%	122.761	73	96%
My Name Is Jonas	212220	315	98%	169.432	161	97%

Os autores do Slashbot relataram dificuldades em identificar a marcação temporal de cada música utilizada para o ajuste dos tempos de espera. Após implementar esta funcionalidade notaram grande perda de desempenho, sendo retirada. Outro aspecto comunicado pelos criadores do robô foi sobre a dificuldade em diferenciar longas sequências de notas consecutivas, das notas contínuas com rastro. Grybot foi construído com identificação de marcação temporal e acerta sequências de notas por meio da implementação do limitador de frequência e captura do rastro.

5.4.2 Grybot vs Cythbot

Grybot utiliza como entrada ao seu sistema de análise de imagem o sinal vídeo composto e realiza o pressionamento dos botões do controle por meio de solenoides. Cythbot obtém a imagem da tela através de uma câmera *Pixelink*[®] e pressiona os botões do controle através de atuadores pneumáticos.

A Tabela 5.19 apresenta o comparativo de desempenho obtido pelos dois robôs no modo *Training*. Considerando esses dados, pode-se concluir que Grybot demonstrou melhor performance que Cythbot, principalmente na música *Through The Fire And Flames*; em que a diferença entre o percentual de acertos entre os dois robôs foi de 17%.

Tabela 5.19: Grybot vs Cythbot [10, 9, 11]

	Grybot		Cythbot	
	Streak	Hits	Streak	Hits
Cult Of Personality	1278	100%	301	99%
Cliffs Of Dover	751	99%	537	99%
Through The Fire and Flames	1496	99%	262	82%

5.4.3 Grybot vs Titan

O robô Titan utiliza captura de imagem por meio de câmera. Apesar da documentação do projeto de construção desse robô não ter sido encontrada, o sistema de pressionamento dos botões da guitarra visto no vídeo analisado aparentou ser constituído por atuadores pneumáticos.

A Tabela 5.20 mostra o desempenho obtido na música *Cliffs of Dover*. Os dados do robô Grybot foram extraídos do modo *Training*, enquanto os dados do Titan foram extraídos do modo *Multiplayer*. Ao contrário do robô Titan, Grybot não utilizou o especial *Star Power*; mas ainda sim, apresentou desempenho superior.

Tabela 5.20: Grybot vs Titan [20]

	Grybot			Titan		
	Score	Streak	Hits	Score	Streak	Hits
Cliffs Of Dover	252.656	751	99%	246.355	565	98%

5.4.4 Grybot vs eGOR

O robô eGOR realiza a captura da tela do jogo através de uma câmera de imagem preto e branco que opera a 30 quadros por segundo. O sinal elétrico de pressionamento dos botões da guitarra são enviados diretamente ao circuito do controle, isto permite que não haja influência de qualquer problema mecânico sobre os resultados alcançados. Porém, dada a necessidade de se desmontar o controle e customizar sua eletrônica, esta solução não pode ser utilizada com qualquer equipamento de videogame, fazendo com que não se possa chama-lo de “robô”, uma vez que não simula ações humanas.

A Tabela 5.21 contém o desempenho alcançado pelos robôs Grybot e eGOR nas músicas *Cult Of Personality* e *Through The Fire And Flames* jogados no modo *Quick Play*. O robô Grybot não dispara o “especial” automaticamente, mas ainda assim foi capaz de atingir marcas melhores do que o eGOR. A diferença entre os resultados foi de 184.141 pontos e 8% de acertos.

Tabela 5.21: Grybot vs eGOR [25, 26]

	Grybot			eGOR		
	Score	Streak	Hits	Score	Streak	Hits
Cult Of Personality	246.458	355	99%	246.304	213	98%
Through The Fire And Flames	669.898	512	99%	485.757	250	91%

5.4.5 Grybot vs DeepNote

A detecção de notas realizada pelo robô DeepNote ocorre por meio do monitoramento do nível de tensão de fotodiodos cuidadosamente posicionados frente a tela do monitor, nos locais que as notas atravessam o braço da guitarra. Assim como no robô anterior, no DeepNote o acionamento das notas é feita pelo envio de sinais elétricos de pressionamento dos botões da guitarra diretamente ao circuito do instrumento.

A Tabela 5.22 apresenta o desempenho dos dois robôs no modo *Quick Play*. DeepNote demonstrou melhor performance comparado ao Grybot, a diferença do entre os percentuais de acertos foi de 0% para a música *Cult Of Personality* e de 1% na música *Through The*

Fire And Flames; a diferença entre as pontuações foi de 66.016 na primeira e 176.642 na segunda. As principais questões que levou Grybot a ficar atrás na pontuação foram: não possuir mecanismo de disparo do *Star Power*, dificuldade em detectar algumas notas com formato de estrela, utilização de um sistema mecânico de pressionamento dos botões do controle (ao invés do envio de sinal direto ao controle). Além disso, a implementação do robô DeepNote envolve um módulo que elimina algumas animações do jogo que afetam o resultado.

Tabela 5.22: Grybot vs DeepNote [6, 5]

	Grybot			DeepNote		
	Score	Streak	Hits	Score	Streak	Hits
Cult Of Personality	246.458	355	99%	312.474	640	99%
Through The Fire And Flames	669.898	512	98%	846.540	N/A	99%

5.4.6 Grybot vs AGH1000

Assim como Grybot, AGH1000 obtém a imagem do jogo através do sinal vídeo composto. Assim como nos dois últimos robôs apresentados, as notas são acionadas por estímulos elétricos introduzidos diretamente no circuito da guitarra. A Tabela 5.23 mostra o desempenho obtido pelos robôs no modo *Quick Play*. A pontuação do AGH1000 foi maior do que a alcançada pelo Grybot, entretanto esse resultado só foi possível devido ao acionamento do *Star Power* e da barra de distorção ter sido realizada por um agente humano.

O robô AGH1000 não conseguiu concluir com sucesso a música *Through The Fire And Flames*.

Tabela 5.23: Grybot vs AGH1000 [24]

	Grybot		AGH1000	
	Score	Hits	Score	Hits
Cliffs Of Dover	221.476	99%	225.123	99%

5.4.7 Grybot vs Seres Humanos

A comparação do desempenho obtido pelo robô frente ao alcançado por pessoas foi realizada tendo por base os resultados cadastrados no site colaborativo ScoreHero. Esta plataforma foi criada em Janeiro de 2006, e permite que jogadores de GH submetam as pontuações atingidas nas diversas músicas e níveis de dificuldade. Para dar confiabilidade aos dados recebidos e divulgados pelo site é necessário enviar provas dos feitos, tais como *screenshots* da tela de resultado mostrado no final de cada partida ou vídeos compartilhados pelo site YouTube.

Analisando a pontuação atingida pelos jogadores do ScoreHero na música *Through The Fire And Flames* (TTFAF) nível *Expert*, constatou-se que Grybot obteve a posição 241st

do rank, com 669.898 pontos e 98% de acertos, enquanto o primeiro colocado (Altruit) ficou com 984.744 pontos e 99% de acertos. A diferença entre as pontuações mostrou ser relativamente grande em comparação com a diferença dos percentuais de acertos, isto ocorreu devido ao robô não possuir mecanismo de disparo do *Star Power*.

Em 2009, o brasileiro de 14 anos Fabio Jardim, se tornou campeão mundial após vencer a competição do *World Cyber Games (WCG)* [31]. Neste mesmo ano, Fábio postou vídeos na internet jogando a música TTFAF no nível *Expert*; alcançou a pontuação 880.862 com 97% de acertos [19].

Também em 2009, Danny Johnson quebrou o record mundial com a maior pontuação realizada na música TTFAF: 985.206 pontos e com 100% de acertos [17], passando a fazer parte do livro dos records.

Capítulo 6

Conclusões

Este trabalho apresentou o desenvolvimento de um robô capaz de jogar as músicas do GH3 em todos os níveis de dificuldade, utilizando uma placa FPGA que recebe o sinal vídeo composto do videogame, detecta a passagem de notas pela tela por meio do monitoramento das cores de pixels da imagem; e envia comandos de acionamento a um circuito de amplificação, que gera corrente suficiente para os solenoides realizarem o pressionamento dos botões do controle.

Considerando os resultados obtidos verificou-se que os objetivos do projeto foram atingidos. O protótipo construído e apresentado neste trabalho foi capaz de concluir todas as músicas do repertório do jogo nos quatro níveis de dificuldade com no mínimo 95% de acertos e demonstrou ótimo desempenho frente a outros robôs desenvolvidos para executar a mesma atividade.

A implementação do reconhecimento de padrões sobre FPGA apresentou-se adequada para promover a análise do sinal de vídeo. Por meio dessa abordagem se tornou possível detectar, com alto nível de precisão, as notas que atravessavam o braço da guitarra.

A adoção do processamento *in line* do sinal VGA RGB reduziu muito o custo computacional quando comparado a soluções que usam visão computacional; permitindo a sua implementação sem a necessidade de usar computadores ou processadores.

A análise realizada com os solenoides utilizados para o pressionamento dos botões do controle mostrou que o sistema mecânico foi eficaz, entretanto muito sensível ao posicionamento e quantidade de acionamentos simultâneos.

Grybot concluiu todas as músicas do jogo GH3 com uma média de acertos de 99,4% no nível *Easy*, 98,8% no nível *Medium*, 98,4% no nível *Hard* e 97,6% no nível *Expert*. Foi um ótimos resultados, pois se aproximaram muito dos que foram alcançados por jogadores especialistas.

Buscando aumentar o desempenho do robô e incorporar novas funcionalidades, propomos os seguintes trabalhos futuros:

- Melhoria da detecção de notas com formato de estrela.
- Aprimorar a análise da imagem buscando diferenciar sequências de notas próximas e rastro de “notas contínuas”.
- Implementar circuito que evite interferência da animação do jogo (raios) sobre a detecção das notas.

- Construção de um sistema mecânico mais robusto.
- Adicionar o acionamento do *Start Power* (especial).
- Adicionar a movimentação do botão analógico (barra de distorção).
- Desenvolver um módulo capaz de jogar no modo *Multiplayer*.
- Adaptar o protótipo para aprender novas velocidades utilizando sistema retroalimentado.

Referências

- [1] Team Titanium FRC 1986. Titan. <http://teamtitanium1986.wordpress.com/robots/titan> [Acesso em 01/07/2013]. x, 15
- [2] Altera. FPGA. <http://www.altera.com/products/fpga.html> [Acesso em 01/07/2013]. 18
- [3] Altera. *DE2 Development and Education Board*, 2007. User Manual. x, xii, 22, 23
- [4] M. de L. S. Batista, P. L. Quintão, S. M. B. Lima, L. C. D. Campos, and T. J. de S. Batista. UM ESTUDO SOBRE A HISTÓRIA DOS JOGOS ELETRÔNICOS. *Revista Eletrônica Faculdade Metodista Granbery*, 2007. 1
- [5] J. Blum. DeepNote GH Bot plays Through the Fire and the Flames. http://www.youtube.com/watch?feature=player_detailpage&v=vyK1uX14LwY [Acesso em 14/09/2013]. xii, 61
- [6] J. Blum. DeepNote GH Bot PROTOTYPE Cult of Personality Expert 99%. http://www.youtube.com/watch?feature=player_detailpage&v=pmzHDdaquk [Acesso em 14/09/2013]. xii, 61
- [7] D. Buckner, M. Jefferis, V. LaPenna, and M. Voth. Slashbot. <http://slashbot.wordpress.com> [Acesso em 10/01/2013]. x, 8, 10
- [8] F. A. C. M Cardoso and M. A. C. Fernandes. FPGA e Fluxo de Projeto. Universidade Estadual de Campinas, 2007. 17, 18
- [9] CythSystemsLLC. Cyth Guitar Hero Cliffs Of Dover Uncut. http://www.youtube.com/watch?feature=player_embedded&v=IY3aMpkWBH8 [Acesso em 14/09/2013]. xii, 59
- [10] CythSystemsLLC. Cyth Guitar Hero Cult of Personality Un-cut. http://www.youtube.com/watch?feature=player_embedded&v=50SEmXDh6Bw [Acesso em 14/09/2013]. xii, 59
- [11] CythSystemsLLC. Cyth Guitar Hero Dragonforce Un-Cut (Surprise). http://www.youtube.com/watch?feature=player_embedded&v=PImO3rp6mXw [Acesso em 14/09/2013]. xii, 59
- [12] K. de O Andrade, G. Fernandes, J. Jr. Martins, V.C. Roma, R.C. Joaquim, and G.A.P. Caurin. Rehabilitation robotics and serious games: An initial architecture

- for simultaneous players. In *Biosignals and Biorobotics Conference (BRC), 2013 ISSNIP*, pages 1–6, 2013. 1
- [13] DesignWorldStaff. Roxanne. <http://www.designworldonline.com/robot-plays-guitar-hero> [Acesso em 01/07/2013]. x, 15, 16
- [14] Doom9. Analog Video. http://www.doom9.org/index.html?/capture/analogue_video.html [Acesso em 14/09/2013]. x, 20
- [15] E. Falk. Guitar Zero. <http://www.kickstarter.com/projects/1695785491/guitar-zero-hardware-mod-for-xbox-360-guitar-contr-0> [Acesso em 01/07/2013]. x, 16
- [16] P. Ganapati. CythBot. <http://www.wired.com/gadgetlab/2008/11/guitar-hero-rob> [Acesso em 15/07/2013]. x, 10
- [17] guinnessworldrecords. HIGHEST SCORE FOR A SINGLE TRACK ON GUITAR HERO III. <http://www.guinnessworldrecords.com/records-6000/highest-score-for-a-single-track-on-guitar-hero-iii/> [Acesso em 14/09/2013]. 62
- [18] National Instruments. National Instruments. <http://www.ni.com> [Acesso em 01/07/2013]. x, 9
- [19] F. Jardim. Guitar Hero 3 - Through The Fire And Flames 880k 97% + Guitar Hero Solo Montage + 100th VIDEO!!! <http://www.youtube.com/watch?v=ckzYVPsNLeE&list=FLINcJGGvKeDHqda2b-vT7oA&index=84> [Acesso em 14/09/2013]. 62
- [20] jspatz1. Team 1986 Guitar Hero Robot - Cliffs of Dover Expert. http://www.youtube.com/watch?feature=player_detailpage&v=0mWHVvKb1hM [Acesso em 14/09/2013]. xii, 60
- [21] D. Lewis and D. Bailey. A checkers playing robot. *Raport instytutowy, Institute of Technology and Engineering, Institute of Information Sciences and Technology Massey University*, 2004. x, 5, 6
- [22] lonyoo. eGOR - Electronic God Of Rock. <http://www.docstoc.com/docs/23881276/eGOR—Electronic-God-Of-Rock> [Acesso em 14/09/2013]. x, 11, 12
- [23] C. Lor and M. Teigen. FVTC Guitar Hero robot. <http://www.fvtc.edu> [Acesso em 01/07/2013]. x, 15
- [24] michaelagh1. Guitar Hero Robot - Cliffs (expert) 225,123. <http://www.youtube.com/watch?v=ITBTUq1WRhg> [Acesso em 14/09/2013]. xii, 61
- [25] mikerrosoftx. eGOR - Guitar Hero Robot - Cult of Personality. http://www.youtube.com/watch?feature=player_detailpage&v=-bonATOIX3g [Acesso em 14/09/2013]. xii, 60

- [26] mikerrosoftx. eGOR - Guitar Hero Robot - Through the Fire and Flames. http://www.youtube.com/watch?feature=player_detailpage&v=xgxK5m9wAQ4 [Acesso em 14/09/2013]. xii, 60
- [27] R. Mizrahi and T. Chalozin. HeroNoid. <http://guitarheronoid.blogspot.com.br> [Acesso em 05/01/2013]. x, 6, 7
- [28] J. P. Nicolle. FPGA. <http://www.fpga4fun.com/> [Acesso em 13/09/2013]. 17
- [29] J. V. Oldfield and R.C. Dorf. *Field-programmable gate arrays: reconfigurable logic for rapid prototyping and implementation of digital systems*. Wiley, 1995. 17
- [30] S. Palnitkar. *Verilog HDL*. Sunsoft Press, 1996. 17
- [31] G. Petró. 'Guitar hero me deu oportunidades' diz brasileiro campeão mundial de games. <http://g1.globo.com/Noticias/Games/0,,MUL1384259-9666,00-GUITAR+HERO+ME+DEU+OPORTUNIDADES+DIZ+BRASILEIRO+CAMP+EO+MUNDIAL+DE+GAMES.html> [Acesso em 14/09/2013]. 62
- [32] ScoreHero. ScoreHero. <http://www.scorehero.com/> [Acesso em 14/09/2013]. 42
- [33] M. Seedman. AGH1000. <http://60035pserver.com/AutoGuitarHeroWebsite> [Acesso em 10/01/2013]. x, 14
- [34] N. Singer, J. Blum, B. Fischer, Z. Lynn, A. Miller, and B. Shaffer. DeepNote. <http://mechanizedrock.com> [Acesso em 10/01/2013]. x, 12, 13
- [35] M. Voth. SlashBot: The Guitar Hero Robot - EXPERT Cliffs of Dover. http://www.youtube.com/watch?feature=player_detailpage&v=icsDptr7Umw [Acesso em 14/09/2013]. xii, 59
- [36] M. Voth. SlashBot: The Guitar Hero Robot - Expert Multiplayer. http://www.youtube.com/watch?feature=player_detailpage&v=OB1qFGJ-8Ww [Acesso em 14/09/2013]. xii, 59
- [37] Wikipedia. O Turco. http://pt.wikipedia.org/wiki/O_Turco. x, 4