



**Universidade Federal de São João del-Rei**  
Campus Alto Paraopeba



## **TRABALHO DE CONTEXTUALIZAÇÃO E INTEGRAÇÃO CURRICULAR – TCIC**

“Implementação de um sistema eletromecânico para auxílio de novos jogadores de xadrez utilizando a plataforma Arduino”

Álvaro Carvalho Vieira  
Erick de Barros Alcântara  
Igor Lima Rodrigues Pereira  
José Antonio Toledo Júnior

Ouro Branco, MG  
Julho de 2018

---

Álvaro Carvalho Vieira  
Erick de Barros Alcântara  
Igor Lima Rodrigues Pereira  
José Antonio Toledo Júnior

**“IMPLEMENTAÇÃO DE UM SISTEMA ELETROMECHANICO PARA  
AUXÍLIO DE NOVOS JOGADORES DE XADREZ UTILIZANDO A  
PLATAFORMA ARDUINO”**

Trabalho apresentado ao Bacharelado Interdisciplinar em Ciência e Tecnologia da Universidade Federal de São João del-Rei (UFSJ), como requisito essencial para obtenção do grau de Bacharel, sob a orientação do Prof. Dr. Guilherme Gomes da Silva.

Ouro Branco, MG

Julho de 2018

## **AGRADECIMENTOS**

É muito bom passar por uma jornada dessas, ter tanto a agradecer e querer homenagear tantas pessoas. É muito valioso dizer obrigado a todos aqueles que, durante este período, em que foi despertado sentimento de tristeza, incapacidade, euforia, incerteza, cansaço, alegrias, conseguiram manter-se inevitavelmente presentes. Por isso, as homenagens:

A Deus, Aquele que se apresentou e se apresenta nas mais variadas formas, sendo este o motivo de continuar a tentar e almejar a conclusão de nossos sonhos, dando-nos força para finalizar este projeto.

Em especial aos nossos pais, a quem devemos a vida, a formação moral e a criação de nossa identidade. Nosso reconhecimento e gratidão por toda a paciência, amor, compreensão e apoio constante.

Aos professores, os parabenizamos pelo compromisso e dedicação. Nossos eternos agradecimentos pelas inúmeras contribuições.

Aos nossos queridos colegas e amigos, pelos momentos de convívio, risos, trocas de conhecimentos, afetos e intrigas, e que por meio delas, amadurecemos.

## RESUMO

O presente trabalho objetiva a elaboração de um tabuleiro de xadrez que auxilie novos jogadores a aprenderem de forma interativa a jogar tal jogo. Para atingir a finalidade proposta, foi utilizado *software* de simulação mecânica, plataforma Arduino, fundamentos teóricos de programação, circuitos elétricos e eletrônica, além das leis que regem o jogo. No desenvolver deste projeto, foi mostrado como *hardware* e *software*, além da estrutura mecânica, foram elaborados para funcionarem em conjunto, obtendo o resultado esperado do protótipo construído para facilitar jogadores novatos de xadrez.

**Palavras Chave:** Projeto, Xadrez, Arduino.

---

## ABSTRACT

The present work aims at the elaboration of a chessboard that helps new players to learn the interactive way of playing the game. For the proposed adaptation, it was used the software of accumulated mechanics, Arduino platform, theoretical fundamentals of programming, electrical and electric, besides the laws that govern the game. In the build this design, it was exposed how hardware and software, in addition to the mechanical structure, were designed to work together, getting the expected result of the prototype to be improved in novices of chess.

**Keywords:** Project, Chess, Arduino.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Tipos de peças do xadrez .....	14
Figura 2 - Posição inicial das peças de xadrez no tabuleiro.....	14
Figura 3 - Possíveis movimentos do bispo .....	15
Figura 4 - Possíveis movimentos da torre .....	15
Figura 5 - Possíveis movimentos da rainha .....	16
Figura 6 - Possíveis movimentos do cavalo .....	16
Figura 7 - Possíveis movimentos do peão .....	17
Figura 8 - Possíveis movimentos do rei .....	17
Figura 9 - Alimentação da placa Arduino Leonardo .....	18
Figura 10 - Circuito regulador para entrada externa.....	19
Figura 11 - Circuito de proteção da USB da placa Arduino Leonardo.....	19
Figura 12 - Circuito de seleção de fonte.....	20
Figura 13 - Conectores de alimentação .....	20
Figura 14 - Microcontrolador ATmega32U4 .....	21
Figura 15 - Pinos de entrada/saída .....	22
Figura 16 - Desenho isométrico da base.....	24
Figura 17 - Desenho isométrico da superfície 2 .....	25
Figura 18 - Desenho isométrico da superfície 1 .....	25
Figura 19 - Desenho isométrico simplificado da barra roscada.....	26
Figura 20 - Desenho isométrico da porca .....	26
Figura 21 - Desenho isométrico da arruela .....	26
Figura 22 - Desenho isométrico do LED .....	27
Figura 23 - Desenho isométrico dos contatos .....	28
Figura 24 - Desenho isométrico do peão .....	28
Figura 25 - Desenho isométrico do cavalo .....	29
Figura 26 - Desenho isométrico do bispo .....	29
Figura 27 - Desenho isométrico da torre .....	29
Figura 28 - Desenho isométrico da rainha .....	30
Figura 29 - Desenho isométrico do rei .....	30
Figura 30 - Desenho isométrico do acabamento xadrez de papel .....	31
Figura 31 - Desenho isométrico da montagem do tabuleiro .....	31

---

Figura 32 - Porta serial do Arduino.....	32
Figura 33 - Representação do circuito de escrita .....	34
Figura 34 - Sub-circuito com DEMUX para o cátodo do circuito de escrita.....	35
Figura 35 - Sub-circuito com DEMUX para o ânodo do circuito de escrita.....	35
Figura 36 - Representação do circuito de leitura.....	37
Figura 37 - Sub-circuito com DEMUX para o circuito de leitura .....	37
Figura 38 - Sub-circuito com MUX para o circuito de leitura .....	38
Figura 39 - Isométrico da montagem física do tabuleiro.....	40
Figura 40 - Visão superior da montagem física do tabuleiro .....	40
Figura 41 - Visão lateral da montagem física do tabuleiro .....	41
Figura 42 - Isométrico do peão físico .....	41

---

## LISTA DE SIGLAS

1. IMSA – Associação Internacional de Esportes Mentais.....	11
2. FIDE – Federação Internacional de Xadrez.....	13
3. USB – <i>Universal Serial Bus</i> .....	18
4. PC – <i>Personal Computer</i> .....	18
5. DC – <i>Direct Current</i> .....	19
6. IOREF – <i>Input/Output Reference</i> .....	20
7. GND – <i>Ground</i> .....	20
8. VIN – <i>Voltage Input</i> .....	21
9. CDC – <i>Communication Device Class</i> .....	21
10. RISC – <i>Reduced Instruction Set Computer</i> .....	21
11. TQFP – <i>Quad Flat Package</i> .....	21
12. SMD – <i>Surface Mount Tecnology</i> .....	21
13. SRAM – <i>Static Random Access Memory</i> .....	21
14. EEPROM – <i>Electrically-Erasable Programmable Read-Only Memory</i> .....	21
15. PWM – <i>Pulse-Width Modulation</i> .....	22
16. RX – <i>Data Reciver</i> .....	22
17. TX – <i>Data Transmitter</i> .....	22
18. I2C – <i>Inter-Integrated Circuit</i> .....	22
19. SDA – <i>Serial Data</i> .....	22
20. SCL – <i>Serial Clock</i> .....	22
21. SPI – <i>Serial Peripheral Interface</i> .....	23
22. ICSP – <i>In-circuit serial programming</i> .....	23
23. AD – <i>Analog-to-Digital</i> .....	23
24. AD – <i>Analog Reference</i> .....	23
25. LED – <i>Light Emitting Diode</i> .....	23
26. DEMUX – <i>Demultiplexer</i> .....	25
27. MUX – <i>Multiplexer</i> .....	26
28. PCB – <i>Printed Circuit Board</i> .....	27
29. CNC – <i>Computer Numeric Control</i> .....	33
30. CD/DVD – <i>Compact Disc / Digital Versatile Disc</i> .....	42



---

## SUMÁRIO

1. INTRODUÇÃO .....	10
2. REFERENCIAL TEÓRICO .....	11
2.1 Xadrez .....	11
2.1.1 Natureza e objetivos do jogo de xadrez .....	13
2.1.2 Posição inicial das peças no tabuleiro .....	13
2.1.3 Movimento das peças .....	14
2.2 Arduino .....	18
2.2.1 Alimentação .....	18
2.2.2 Comunicação USB .....	21
2.2.3 Microcontrolador .....	21
2.2.4 Pinos de entrada e saída .....	22
3. OBJETIVOS .....	23
4. MÉTODOS E RESULTADOS .....	24
4.1 Mecânica .....	24
4.1.1 Desenho das peças .....	28
4.1.2 Resultado da montagem no <i>software</i> .....	30
4.2 Programação .....	31
4.3 Eletrônica .....	33
4.3.1 Circuito de escrita .....	33
4.3.2 Circuito de leitura .....	36
4.4 Resultados da Montagem Física Completa .....	38
5. CONCLUSÃO .....	42
6. REFERÊNCIAS .....	43

## 1. INTRODUÇÃO

Ao contextualizar o termo automação, tem-se como foco e visão um sistema automático de controle pelo qual os mecanismos verificam seu próprio funcionamento, efetuando medições e introduzindo correções, sem a necessidade da interferência do homem [1].

Em seu uso moderno, a automação pode ser definida como uma tecnologia que utiliza comandos programados para operar um dado processo, combinados com retroação de informação para determinar que os comandos sejam executados corretamente. Frequentemente utilizada em processos antes operados por seres humanos, é a aplicação de técnicas computadorizadas ou mecânicas para diminuir o uso de mão-de-obra em qualquer processo [2].

Ao ingressar no curso de Engenharia Mecatrônica na Universidade Federal de São João del-Rei - Campus Alto Paraopeba (UFSJ-CAP), são apresentados ao aluno conhecimentos na área da automação, controle, eletrônica, programação e mecânica durante aulas, palestras, seminários, viagens técnicas, feiras e outras atividades complementares que possibilitam a formação de uma pessoa dinâmica, criativa, lógica e resiliente.

Por meio deste processo o aluno está capacitado a procurar oportunidades de projetos, conceber a ideia e desenvolvê-las por meio de uma Iniciação Científica, TCC, TCIC ou Programa de Extensão. Tudo isso é estruturado por uma extensa pesquisa, parte primordial para a implementação de qualquer projeto.

A popularização dos microcontroladores junto com a queda dos custos de componentes eletrônicos, baterias e motores, possibilitaram o desenvolvimento de pequenas plataformas autônomas de baixo custo [3].

A implementação de uma plataforma automática envolve conceitos de eletrônica, computação e mecânica que, integrados de forma sincronizada, possibilitam a construção de um sistema embarcado. Para realizar essa integração é utilizado um microcontrolador que gerenciam os periféricos a ele conectados. Através de embasamentos teóricos, ocorre a ideia de criar um “xadrez interativo”, com o

objetivo de auxiliar novos jogadores no processo de aprendizado sobre o jogo, tornando mais intuitivo e ágil esse processo.

## **2. REFERENCIAL TEÓRICO**

### **2.1 Xadrez**

O jogo de xadrez, integrado como esporte mental pela Associação Internacional de Esportes Mentais (IMSA) desde 2005, é reconhecido por desenvolver as capacidades de cognição como memória, tomada de decisão, raciocínio lógico-matemático e inteligência espacial.

Segundo o papa Inocêncio III (papa entre 1198 e 1216) [4]:

Este mundo todo é como um tabuleiro de xadrez: uma casa é branca, outra casa é preta, e assim representa o duplo estado de vida ou de morte, de graça ou pecado. A família que habita esse tabuleiro é formada pelos homens deste mundo, que – tal como as peças saídas todas da mesma bolsa – procedem todos de um só ventre materno. E, tal como as peças, assumem seus postos nos diferentes lugares deste mundo, cada um com sua própria denominação. O primeiro é o Rei, depois a Rainha, em terceiro lugar a Torre, em quarto o Cavalo, em quinto o Bispo e em sexto o Peão. E o caráter do jogo é tal que um toma o outro e, com o jogo terminado, assim como todos tinham saído da mesma bolsa, a ela voltam. E então já não há diferença entre o Rei e o pobre Peão, pois acabam do mesmo modo o rico e o pobre. E com frequência acontece que, quando se devolvem as peças, o Rei fica por baixo, no fundo do saco; e assim também acontece com os grandes que ao sair deste mundo são sepultados no inferno; enquanto os pobres são levados ao seio de Abraão.

A referência [5] traz as informações a seguir, sendo que para eles o xadrez apresenta-se como uma rica ferramenta no processo de tomada de decisões para os diversos fatores de nossa vida. Diferentemente de outros esportes-jogos, como no futebol onde o centroavante desperdiça vários chutes à gol – concluindo muitas vezes apenas uma tentativa entre tantas – o xadrez com toda a sua precisão não permite este “luxo”, pois qualquer que seja pequeno o erro de cálculo ou má visualização da tomada de decisão, pode acarretar em uma derrota imediata, assim:

“As decisões são escolhas tomadas com base em propósitos, são ações orientadas para determinado objetivo e o alcance deste objetivo determina a eficiência do processo de tomada de decisão” (SHENK, 2006).

Xadrez é claramente um "caminho ideal para o estudo de tomada de decisão e resolução de problemas, porque é um sistema fechado, com regras claramente definidas" (Horgan, 1988).

Quando confrontados com um problema, o primeiro passo é "analisar [o] de uma forma preliminar e impressionista: avaliando o problema" (Horgan, 1988, p. 3), possivelmente à procura de padrões ou semelhança com experiências anteriores. "Juízos de similaridade pode envolver altos níveis de raciocínio abstrato" (Horgan, 1988, p. 3). Como o estudo de padrões na matemática, o reconhecimento de padrões em xadrez é de primordial importância na resolução de problemas. Depois de reconhecer a semelhança de padrões, uma estratégia global pode ser desenvolvida para resolver o problema. Trata-se de alternativas de produção, um processo criativo.

Um bom jogador de xadrez, como um solucionador de problemas bom, tem "adquirido um vasto número de esquemas inter-relacionados" (Horgan, 1988, p. 3), permitindo uma boa alternativa para rápida e fácil resolução vêm à mente. Essas alternativas devem ser avaliadas, através de um processo de cálculo conhecido como análise da árvore de decisão, onde solver o jogador de xadrez / problema é calcular a conveniência de eventos futuros com base nas alternativas estão sendo analisadas.

Horgan (1988) constatou que "o cálculo pode ir várias jogadas a oito ou dez anos à frente. Esta fase exige concentração grave e habilidades de memória ... [ou] ... imagética visual" (p.4).

Depois que uma alternativa adequada para resolver o problema é alcançada e implementada, ela pode ser avaliada. Os jogadores de xadrez, como todos os bons solucionadores de problemas, voltam e avaliam o resultado de uma solução para aumentar o seu nível de especialização. "Peritos e especialistas em potencial quer saber, mesmo quando eles forem bem-sucedidos, se houvesse uma alternativa melhor à sua disposição" (Horgan 1988, p.6). Segundo Bloom (1956), este processo de avaliação é um dos objetivos mais importantes da aprendizagem e deve, portanto, ser considerado um dos maiores objetivos educacionais de nossas escolas. "A tendência do xadrez para desenvolver habilidades que podem ser usadas para lidar com as complexidades da vida o tornam uma ferramenta valiosa para o aprendizado.

Xadrez deve ser uma disciplina optativa no currículo das escolas públicas" (Schmidt, p. 6).

### **2.1.1 Natureza e objetivos do jogo de xadrez**

Segundo as leis da Federação Internacional de Xadrez (FIDE 2009), o jogo de xadrez é disputado entre dois oponentes que movem peças alternadamente sobre um tabuleiro quadrado denominado tabuleiro de xadrez. O jogador com as peças brancas começa o jogo. Diz-se que um jogador "tem a vez de jogar" quando a jogada do seu oponente tiver sido feita.

O objetivo de cada jogador é colocar o rei do oponente sob ataque de tal forma que o oponente não tenha lance legal. O jogador que alcançar esse objetivo diz-se que deu "xeque-mate" no rei do adversário e venceu a partida. Não é permitido deixar ou colocar o seu próprio rei sob ataque, bem como capturar o rei do oponente.

O oponente cujo rei sofreu xeque-mate perdeu a partida. A partida está empatada se resultar numa posição em que nenhum dos jogadores tem possibilidade de dar xeque-mate [6].

### **2.1.2 Posição inicial das peças no tabuleiro**

No início da partida, um jogador tem 16 peças de cores brancas e o outro tem 16 peças de cores pretas, sendo o tipo e quantidade de cada peça demonstrados na Figura 1.

O tabuleiro de xadrez é composto de uma rede de 8x8 com 64 casas iguais, alternadamente entre claras (brancas) e escuras (pretas), como mostrado na Figura 2. O tabuleiro é colocado entre os jogadores de tal forma que seja branca a casa do canto à direita de cada jogador.

Peça	Representação Gráfica	Quantidade por Jogador
Rei	 ou 	1
Dama	 ou 	1
Torre	 ou 	2
Bispo	 ou 	2
Cavalo	 ou 	2
Peão	 ou 	8

**Figura 1 - Tipos de peças do xadrez [6]**



**Figura 2 - Posição inicial das peças de xadrez no tabuleiro [6]**

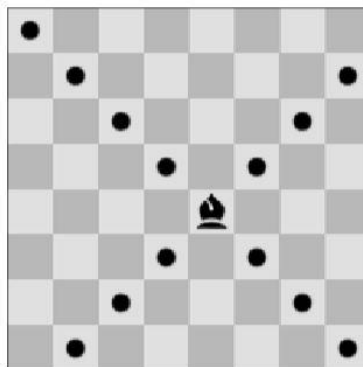
As oito casas dispostas verticalmente são chamadas de colunas. As oito casas dispostas horizontalmente são chamadas de fileiras. A linha reta de casas da mesma cor, movendo-se de uma ponta do tabuleiro a uma ponta adjacente, é chamada de diagonal [6].

### 2.1.3 Movimento das peças

Não é permitido mover uma peça para uma casa já ocupada por outra peça de mesma cor. Se uma peça for movida para uma casa já ocupada por uma peça do oponente, esta última é capturada e retirada do tabuleiro como parte do mesmo movimento. Diz-se que uma peça está atacando uma peça do oponente se puder efetuar uma captura naquela casa.

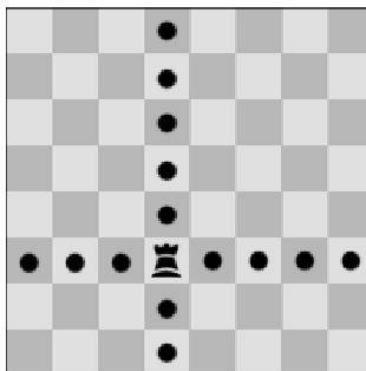
Considera-se que uma peça ataca uma casa, mesmo se tal peça está impedida de ser movida para esta casa, porque, conseqüentemente, deixaria ou colocaria o seu próprio rei sob ataque.

O bispo pode mover-se para qualquer casa ao longo da diagonal em que se encontra, como mostrado na Figura 3.



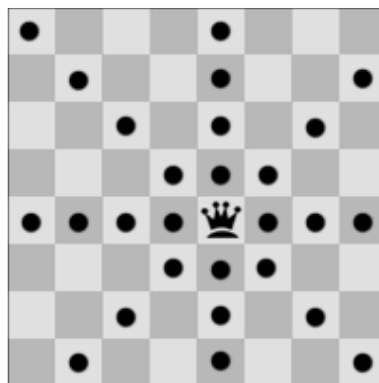
**Figura 3 - Possíveis movimentos do bispo [6]**

A torre se movimenta para frente e para trás, para a direita e para a esquerda, quantas casas quiser, como mostrado na Figura 4.



**Figura 4 - Possíveis movimentos da torre [6]**

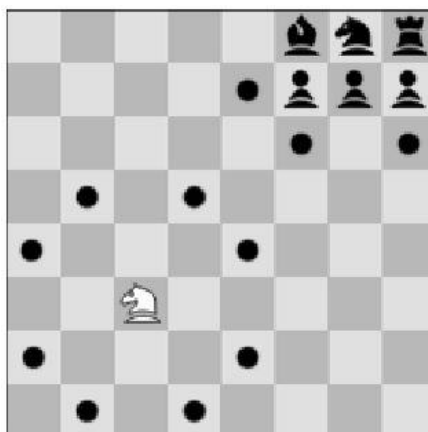
A dama pode mover-se para qualquer casa ao longo da coluna, fileira ou diagonal em que se encontra, como mostrado na Figura 5.



**Figura 5 - Possíveis movimentos da rainha [6]**

Ao fazer esses movimentos, o bispo, torre ou dama não podem pular sobre qualquer peça que esteja em seu caminho.

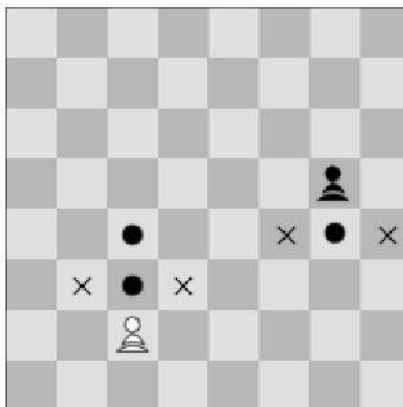
O cavalo pode ser movido para uma das casas mais próxima em relação à qual ocupa, como mostrado na Figura 6. Note que a casa para a qual o cavalo pode mover-se não está na mesma coluna, fileira ou diagonal.



**Figura 6 - Possíveis movimentos do cavalo [6]**

O peão pode ser movido para uma casa, imediatamente à sua frente, na mesma coluna, que não se encontre ocupada, ou em seu primeiro lance o peão pode ser movido conforme mencionado; alternativamente pode avançar duas casas ao longo da mesma coluna, desde que ambas estejam desocupadas, ou o peão pode ser movido para uma casa ocupada por uma peça do oponente, que esteja diagonalmente à sua frente, numa coluna adjacente, capturando aquela peça. É possível perceber tais movimentações na Figura 7.

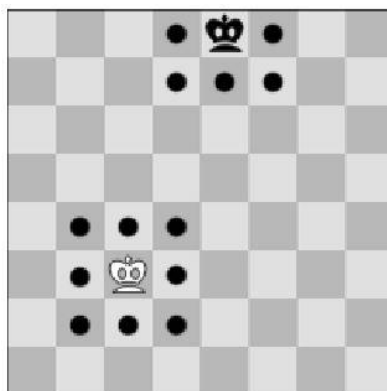




**Figura 7 - Possíveis movimentos do peão [6]**

Um peão atacando uma casa atravessada pelo peão do oponente que acaba de avançar duas casas num único lance vindo de sua casa original, pode capturar este peão oponente como se aquele tivesse se movido apenas uma casa. Esta captura pode ser feita apenas no movimento imediatamente após o referido avanço e é chamada de tomada '*en passant*'.

Há duas formas diferentes de mover o rei: movendo-se o rei para qualquer casa vizinha não atacada por uma ou mais peças do oponente ou '*rocando*'. Este é um lance efetuado com o rei e uma das torres, de mesma cor ao longo da primeira fileira do jogador, considerado como um único lance de rei e executado da seguinte forma: o rei é transferido de sua casa original a duas casas em direção à torre que está em sua casa original, em seguida a torre é transferida para a casa a que o rei acabou de atravessar [6]. O movimento possível do rei está apresentado na Figura 8.



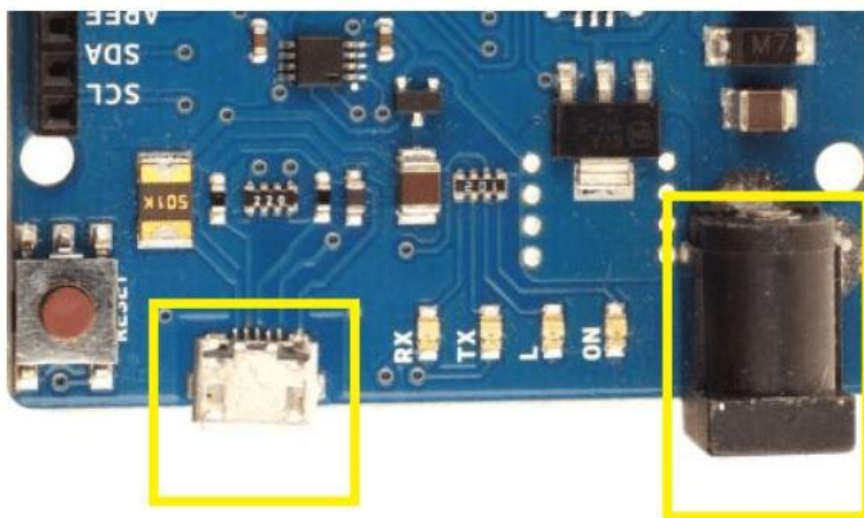
**Figura 8 - Possíveis movimentos do rei [6]**

## 2.2 Arduino

A plataforma aberta Arduino [7] possui interface intuitiva, preço acessível e é utilizada bastante para testes ou fins educacionais. O Arduino Leonardo faz parte do portfólio de placas disponíveis no mercado. Ele difere de placas Arduino anteriores, pois possui um microcontrolador ATMELE de 8-bits com o periférico USB, o que elimina o uso de um segundo microcontrolador com USB para programação. Além disso, essa USB pode ser usada para controle pelo PC [8].

### 2.2.1 Alimentação

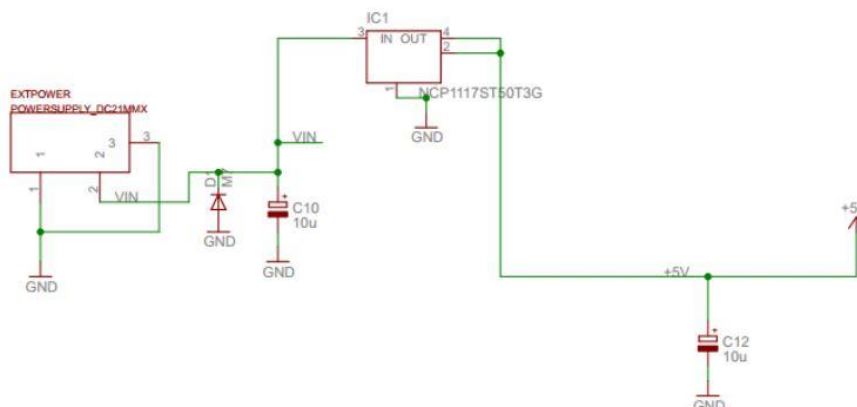
O Arduino Leonardo, assim como o Arduino UNO, pode ser alimentado tanto pelo USB quanto por uma fonte externa, mostrada na Figura 9.



**Figura 9 - Alimentação da placa Arduino Leonardo [8]**

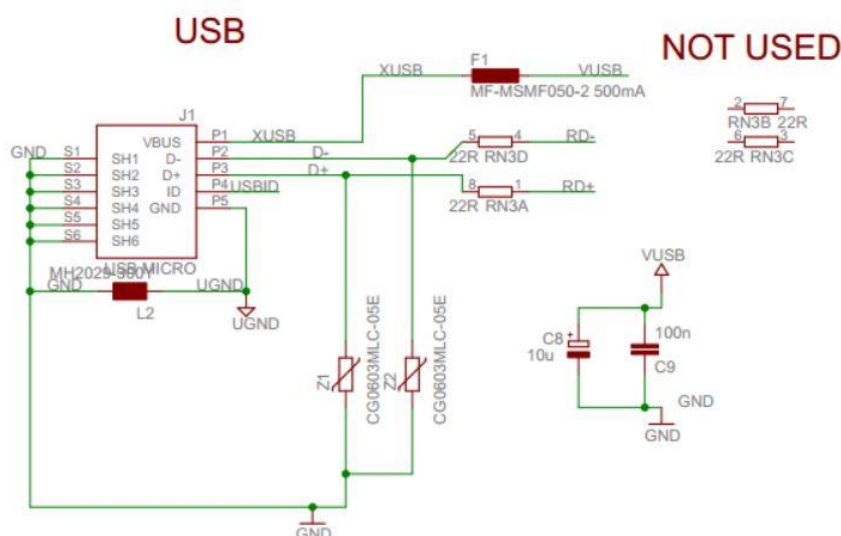
Como no Arduino UNO, a alimentação externa é feita através do conector Jack com positivo no centro, onde o valor de tensão da fonte externa deve estar entre 6 V e 20 V. Se alimentada com uma tensão abaixo de 7 V para o UNO e abaixo de 5 V para o Leonardo, que são suas respectivas tensões de funcionamento da placa, pode haver instabilidade e quando alimentada com tensão acima de 12 V, o regulador de tensão da placa pode sobreaquecê-la e danificá-la.

O circuito regulador para entrada externa é exibido na Figura 10.



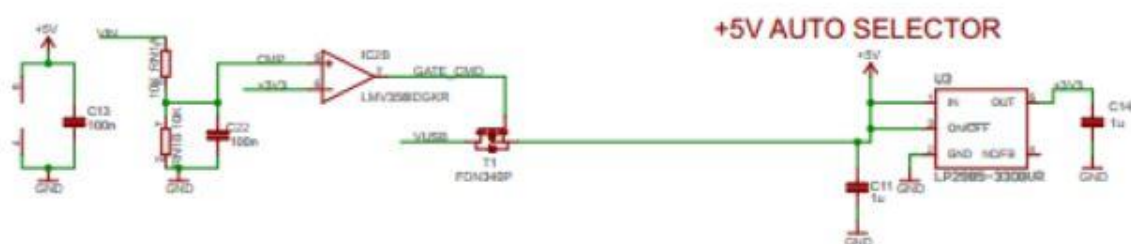
**Figura 10 - Circuito regulador para entrada externa [8]**

Quando o Arduino Leonardo é alimentado pela USB, a tensão da porta USB não precisa ser estabilizada pelo regulador de tensão. Dessa forma a placa é alimentada diretamente pelo USB. O circuito do USB exibido na Figura 11 apresenta alguns componentes que protegem a porta USB do computador em caso de alguma anormalidade.



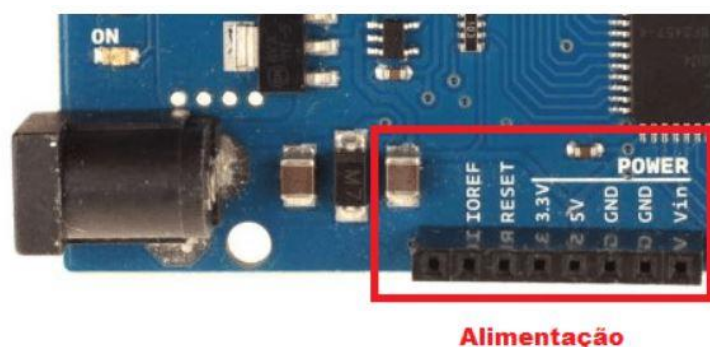
**Figura 11 - Circuito de proteção da USB da placa Arduino Leonardo [8]**

O Arduino Leonardo possui o circuito mostrado na Figura 12 para seleção da fonte de alimentação e alterna automaticamente entre a tensão do USB e a tensão da fonte externa. Caso haja uma tensão no conector DC e o USB é conectado, a tensão de 5V será proveniente da fonte externa e USB servirá apenas para comunicação com o computador.



**Figura 12 - Circuito de seleção de fonte [8]**

Na Figura 13 são exibidos os conectores de alimentação para conexão de *shields* e módulos na placa Arduino Leonardo [8].



**Figura 13 - Conectores de alimentação [8]**

**IOREF** – Fornece uma tensão de referência para que *shields* possam selecionar o tipo de interface apropriada. Dessa forma *shields* que funcionam com as placas Arduino que são alimentadas com 3,3V podem se adaptar para ser utilizados em 5V e vice-versa.

**RESET** – Pino conectado a pino de *RESET* do microcontrolador. Pode ser utilizado para um reset externo da placa Arduino.

**3,3V** – Fornece tensão de 3,3V para alimentação de *shields* e módulos externos. Corrente máxima de 50mA.

**5V** – Fornece tensão de 5V para alimentação de *shields* e circuitos externos.

**GND** – Pinos de referência, terra.

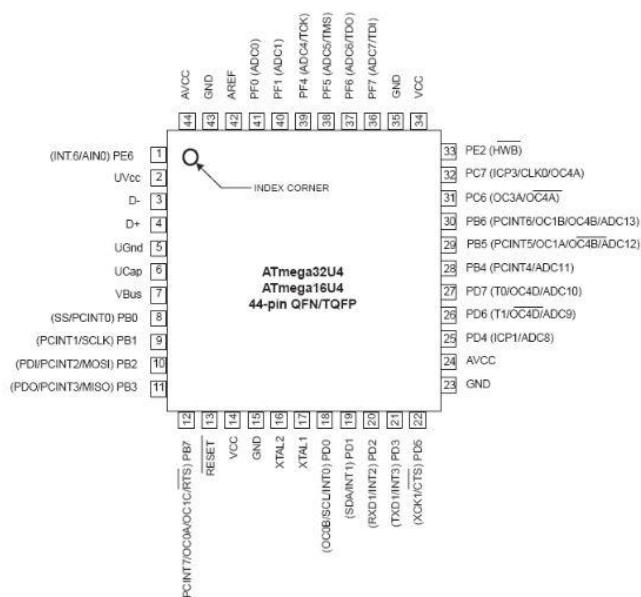
**VIN** – Pino para alimentar a placa através de *shield* ou bateria externa. Quando a placa é alimentada através do conector Jack, a tensão da fonte estará nesse pino.

## 2.2.2 Comunicação USB

A placa Arduino Leonardo não possui circuito para interface USB, como o Arduino Uno. A comunicação USB é feita pelo mesmo microcontrolador que executa o *sketch* compilado. O microcontrolador Atmega32U4, que é o protagonista dessa placa, possui comunicação USB integrada, assim não há a necessidade de um conversor USB serial, como ocorre em outras placas Arduino. Este recurso reduz o custo da placa. O USB pode operar como CDC *device*. Isso permite emular um teclado, *mouse*, *joystick* ou qualquer dispositivo CDC padrão de mercado. O *device* USB dele é 2.0, ou seja, pode atingir velocidade de comunicação de até 12MB/s [8].

## 2.2.3 Microcontrolador

O microcontrolador utilizado no Arduino Leonardo é o ATmega32U4, apresentado na Figura 14, de 8 bits da família AVR com arquitetura RISC avançada. Possui encapsulamento TQFP que apenas permite a montagem em superfície, ou seja, é um componente SMD. Possui 32KB de memória Flash (porém utiliza 4KB para o bootloader), 2,5KB de SRAM e 1KB de EEPROM [8].



**Figura 14 - Microcontrolador ATmega32U4 [8]**

## 2.2.4 Pinos de entrada e saída

A placa Arduino Leonardo possui pinos de entradas e saídas digitais, assim como pinos para saída PWM e pinos para entradas analógicas. Possui pinagem no padrão REV3, conforme exibido na Figura 15.

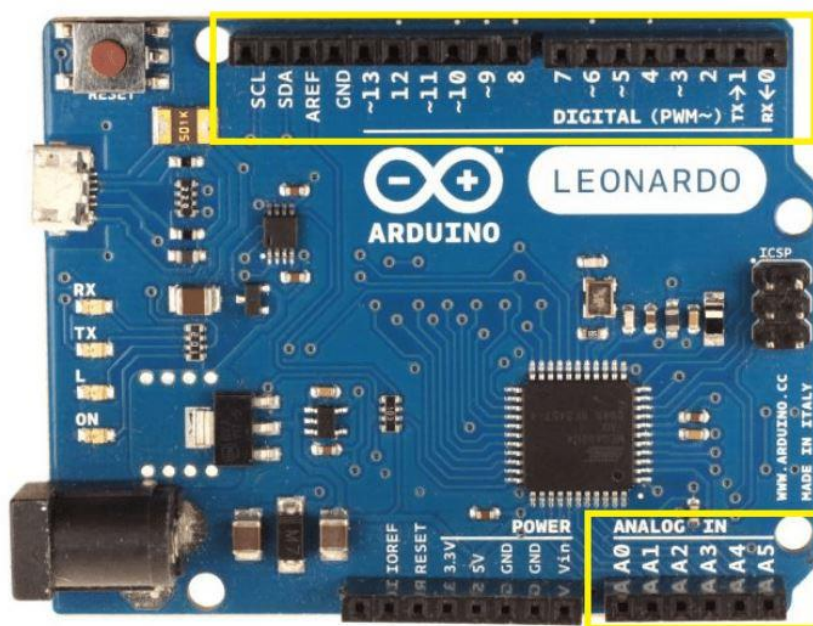


Figura 15 - Pinos de entrada/saída [8]

Todos os pinos podem ser usados como entradas ou saídas digitais, inclusive os que estão dispostos para entradas analógicas. Estes pinos operam em 5V, onde cada pino pode fornecer ou receber uma corrente máxima de 40mA. Cada pino possui resistor de *pull-up* interno que pode ser habilitado por *software*. Esse resistor tem o valor entre 20 e 50K $\Omega$  [8].

**Comunicação Serial** – pinos 0 (RX) e 1 (TX). Esses pinos são usados para enviar e receber dados através de comunicação serial. É interessante notar que para usar esses pinos deve-se usar a classe Serial1, pois a classe Serial refere-se à comunicação USB (CDC).

**Comunicação TWI (I2C)** – pinos 2 (SDA) e 3 (SCL). Permitem a comunicação TWI (I2C da Atmel) usando a biblioteca Wire.



**Interrupção externa** – pinos 3 (*interrupt 0*), 2 (*interrupt 1*), 0 (*interrupt 2*), 1 (*interrupt 3*) e 7 (*interrupt 4*). Esses pinos podem ser configurados para disparar uma interrupção quando ocorrer uma mudança no estado, uma borda de descida ou subida, ou um nível baixo. Para mais detalhes verifique a função *attachInterrupt()*.

**Saídas PWM** – pinos 3, 5, 6, 9, 10, 11 e 13. Podem ser utilizados como saídas PWM de 8 bits de resolução através da função *analogWrite()*.

**Comunicação SPI** – apenas está disponível no conector ICSP e pode ser utilizado com a biblioteca SPI.

**Entradas Analógicas** – para interfacear com o mundo analógico, a Arduino Leonardo possui 12 entradas analógicas. As entradas conhecidas para esse propósito são A0-A5, além de mais 6 entradas que estão no lado dos pinos digitais, que são denominadas A6 a A11 e estão respectivamente nos pinos digitais 4, 6, 8, 9, 10 e 12. O conversor AD do ATmega32U4 possui resolução de 10 *bits* e sua referência está ligada internamente a 5V, ou seja, quando a entrada estiver com 5V o valor da conversão analógica digital será 1023. O valor da referência pode ser mudado através do pino AREF.

### 3. OBJETIVOS

O principal objetivo desse projeto é a elaboração de um tabuleiro de xadrez que seja capaz de mapear o jogo via *software* e indicar fisicamente através de luzes no próprio tabuleiro quais posições a peça que se deseja mover poderá assumir. O projeto em questão tem como objetivos específicos:

- I. Utilizar uma interface que permita ao usuário entrar com a posição da peça que se deseja mover;
- II. Implementar um circuito eletrônico que possibilite acender qualquer um dos 64 *LEDs* do tabuleiro, a fim de mostrar ao usuário onde o mesmo pode mover a peça. Este circuito será denominado de “circuito de escrita”;
- III. Desenvolver uma lógica computacional para analisar as jogadas, recebendo a posição da peça que se deseja mover e acendendo os *LEDs* corretos para aquela jogada do usuário.

Como possível extensão do projeto, tentar-se-á elaborar um circuito que identifique automaticamente qual peça será movida simplesmente levantando-a do tabuleiro, eliminando assim a necessidade de indicar via *software* qual será a peça. Este circuito será denominado de “circuito de leitura”.

## 4. MÉTODOS E RESULTADOS

Para uma melhor explicação, o conteúdo desta seção foi dividido em 3 subseções (Mecânica, Programação e Eletrônica) intimamente relacionadas, porém individualmente explicadas.

### 4.1 Mecânica

Pensando na maior interatividade possível entre homem e equipamento, primeiramente implementou-se no SolidWorks [9] um projeto mecânico que auxiliasse os alunos na concepção do mesmo, tornando assim mais palpável e de fácil visualização de possíveis desafios durante montagem.

Foram utilizadas 3 placas de madeira 300x300x6mm de dimensão. A placa mais em baixo representada na Figura 16 será denominada “base”, enquanto do centro representada na Figura 17 será nomeada “superfície 2” e, por fim, a superior representada na Figura 18 será chamada de “superfície 1”. A base comporta o Arduino e as placas eletrônicas, ficando distante das superfícies, que contêm o encaixe dos *LEDs*, peças e contatos.

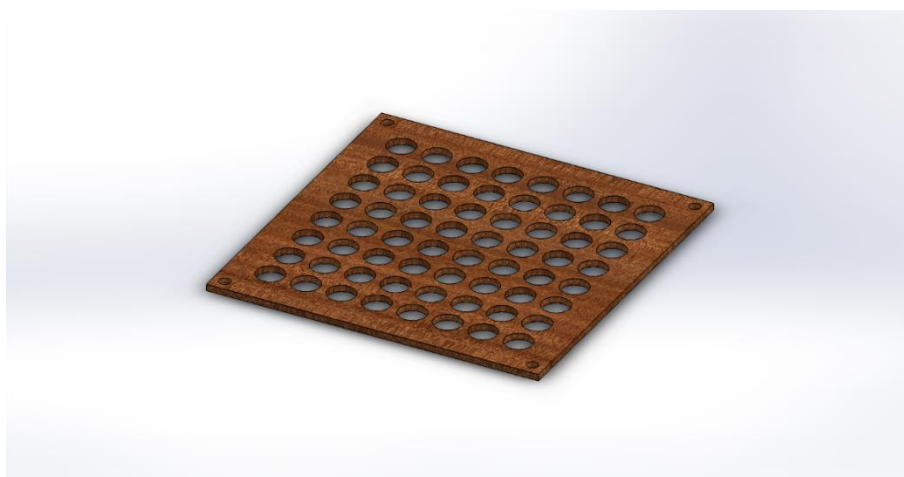


**Figura 16 - Desenho isométrico da base**



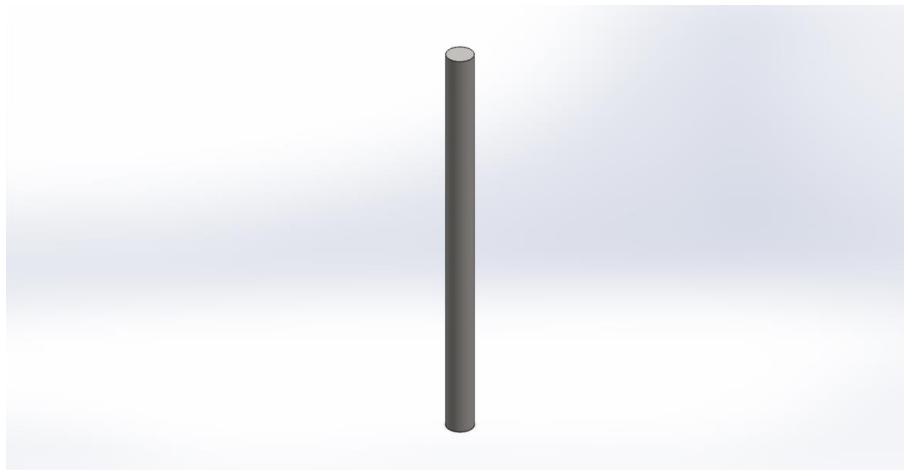


**Figura 17 - Desenho isométrico da superfície 2**

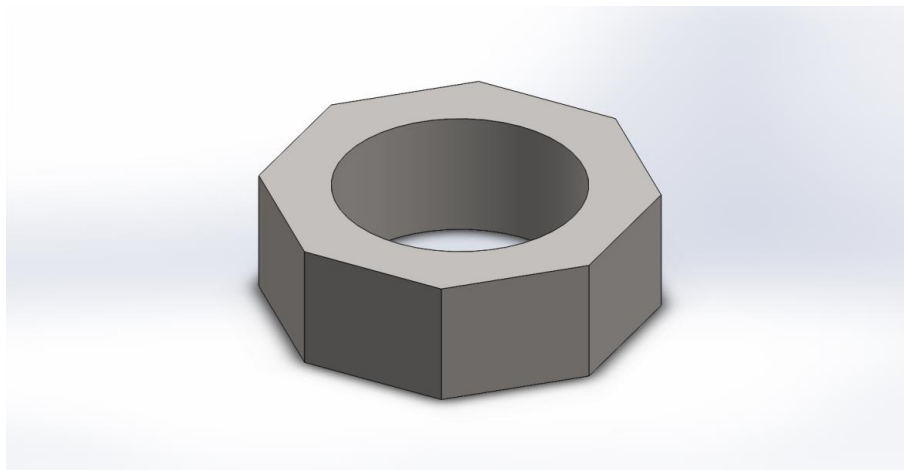


**Figura 18 - Desenho isométrico da superfície 1**

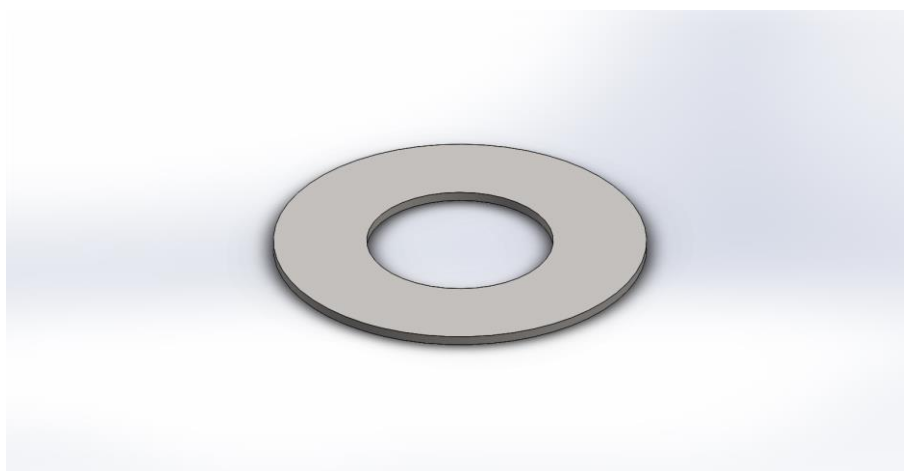
A fim dos circuitos e dos fios ficarem no meio da base e das superfícies permitindo fácil manuseio dos mesmos, um sistema de fixação composto por 4 barras roscadas de 150mm de altura e 3/8" de diâmetro, 16 porcas sextavadas e 16 arruelas foi utilizado, apresentados na Figura 19, Figura 20 e Figura 21, respectivamente. Para uma boa fixação mecânica, passou-se para cada furo de sustentação uma arruela na parte de dentro e na parte externa na chapa de madeira e o fechamento deu-se através de duas porcas.



**Figura 19 - Desenho isométrico simplificado da barra roscada**



**Figura 20 - Desenho isométrico da porca**



**Figura 21 - Desenho isométrico da arruela**

A base tem somente o furo de fixação, porém as outras duas placas de madeira tiveram furações diferentes, pois, além dos furos de fixação, por elas passaram os

*LEDs*, os contatos e os furos maiores, necessário para encaixar as peças. Um desenho simplificado de *LED* está exibido na Figura 22.

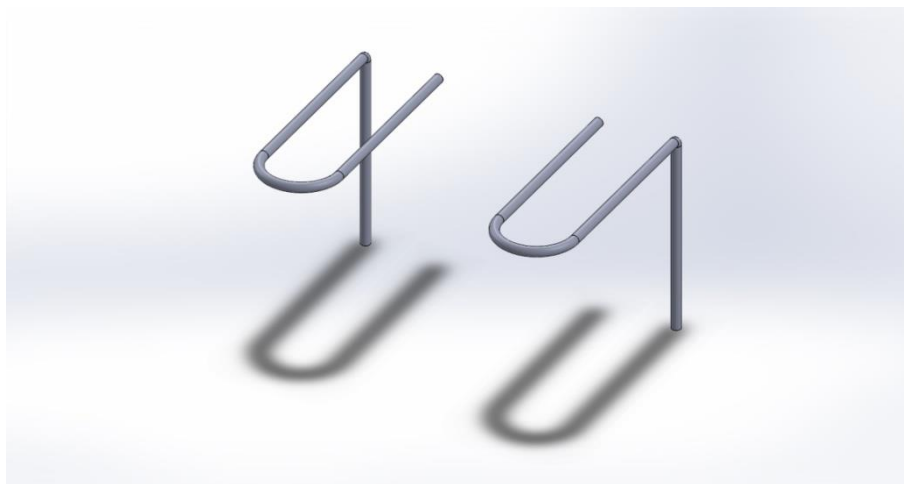


**Figura 22 - Desenho isométrico do LED**

Em específico a superfície 1 apresenta os furos de encaixe das peças e dos *LEDs*, já a superfície 2 possui os furos dos contatos e também dos *LEDs*. Essa “camada” superficial foi dividida em duas placas devido a facilidade do *CNC* em perfurar por completo uma das placas de 6mm ao invés de perfurar e remover um cilindro de 6mm de altura e 25mm de diâmetro numa chapa de 12mm inteiriça.

Os furos dos *LEDs* e contatos foram feitos na *CNC* com broca de diâmetro 2mm, mas a espessura do fio é de aproximadamente 0,5mm, o que proporcionou uma folga para fácil movimentação, caso fosse necessário. Como a superfície 2 contém apenas furos muito pequenos, exceto os de sustentação, a imagem da Figura 17 assemelha-se muito com a base mostrada na Figura 16, porém se for dado o zoom é possível ver que a base não contém os furos de 2mm que a superfície 2 contém.

Os contatos foram feitos através de fios, devido ao seu baixo custo, fácil manuseio e montagem facilitada quando comparados ao uso de pequenas chapas metálicas, a qual deveria ser cortada e soldada aos fios da placa de escrita, o que demandaria mais trabalho e mais gastos financeiros. Um exemplo de par de contatos está mostrado na Figura 23. Antes da montagem física passou-se papel alumínio nos fios para que obtivesse uma maior área de contato.

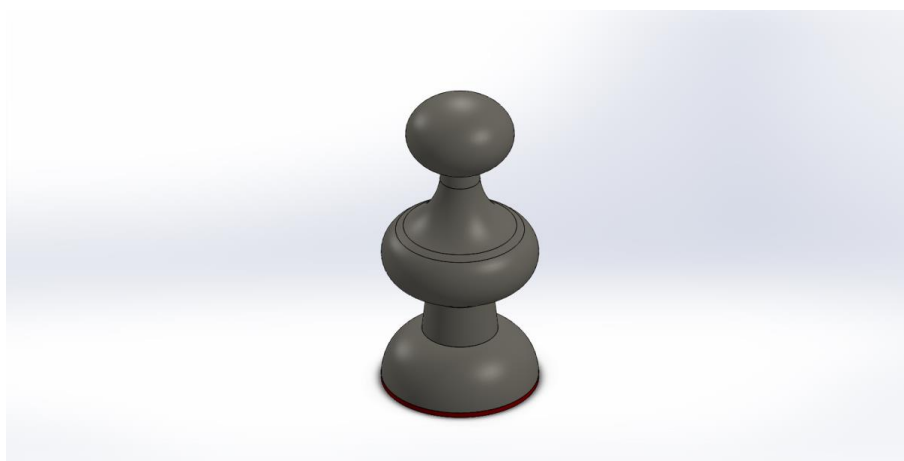


**Figura 23 - Desenho isométrico dos contatos**

Uma solução de melhor implementação e mais alta confiabilidade seria o uso de *Reed Switchs* ou sensores de fim de curso, porém como seriam 64 componentes desses, o valor do projeto sairia do previsto, descartando assim essa opção.

#### **4.1.1 Desenho das peças**

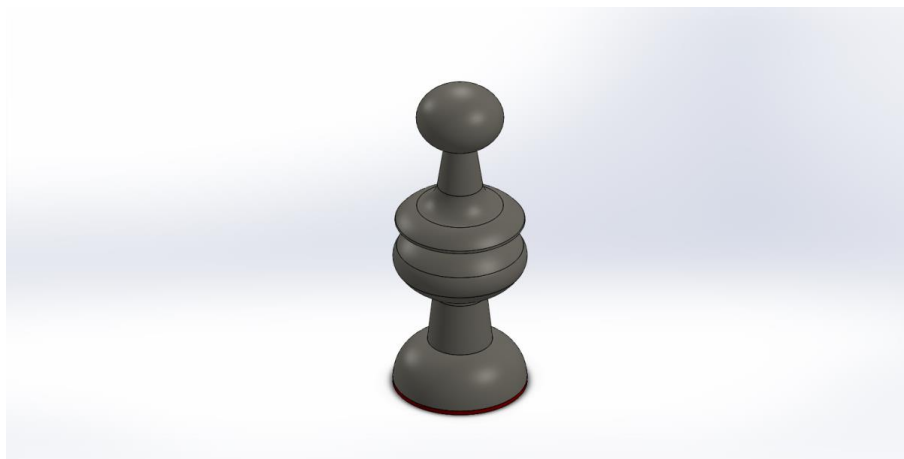
As peças do tabuleiro de xadrez foram retiradas de um projeto *Open Source* da internet [10], sendo necessário apenas a adaptação das medidas de cada peça para adequação ao projeto. As peças apresentadas são da cor preta, porém há também as de cor branca que não foram exibidas isoladamente aqui.



**Figura 24 - Desenho isométrico do peão [10]**



**Figura 25 - Desenho isométrico do cavalo [10]**



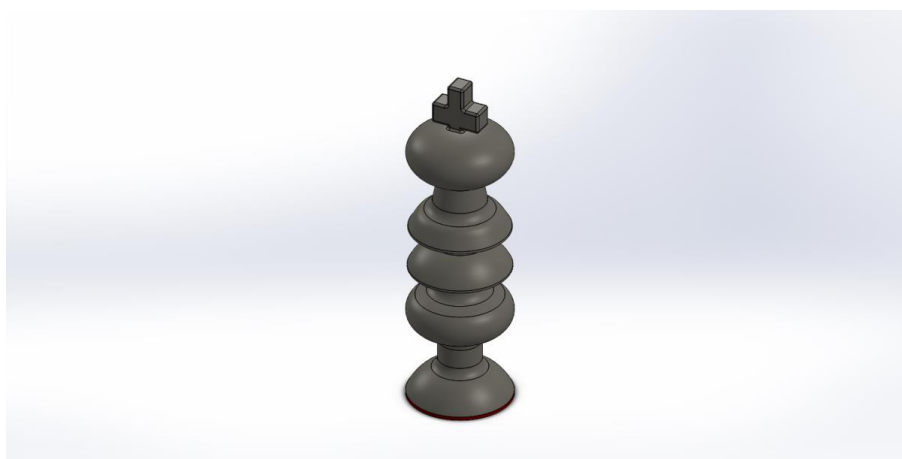
**Figura 26 - Desenho isométrico do bispo [10]**



**Figura 27 - Desenho isométrico da torre [10]**



**Figura 28 - Desenho isométrico da rainha [10]**

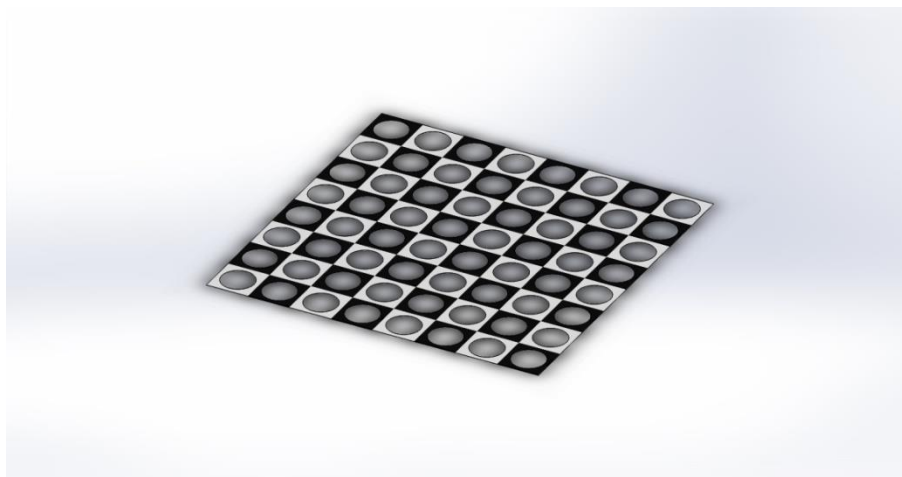


**Figura 29 - Desenho isométrico do rei [10]**

#### **4.1.2 Resultado da montagem no *software***

Antes da obtenção da montagem final, colou-se 2 arruelas na base de cada peça do tabuleiro para dar mais aderência e peso, a fim de melhorar o contato com o papel alumínio – foi necessário lixar o lado de contato da arruela com o papel alumínio.

Um papel xadrez foi adicionado à montagem para a estética do projeto. Este papel e o resultado da montagem no SolidWorks podem ser conferidos na Figura 30 e Figura 31, respectivamente.



**Figura 30 - Desenho isométrico do acabamento xadrez de papel**



**Figura 31 - Desenho isométrico da montagem do tabuleiro**

## **4.2 Programação**

O código disponível na referência [11] foi criado na plataforma Arduino e permite controlar os pinos do microcontrolador utilizado a fim de acender os *LEDs* nas posições corretas para todas as peças e situações do jogo.

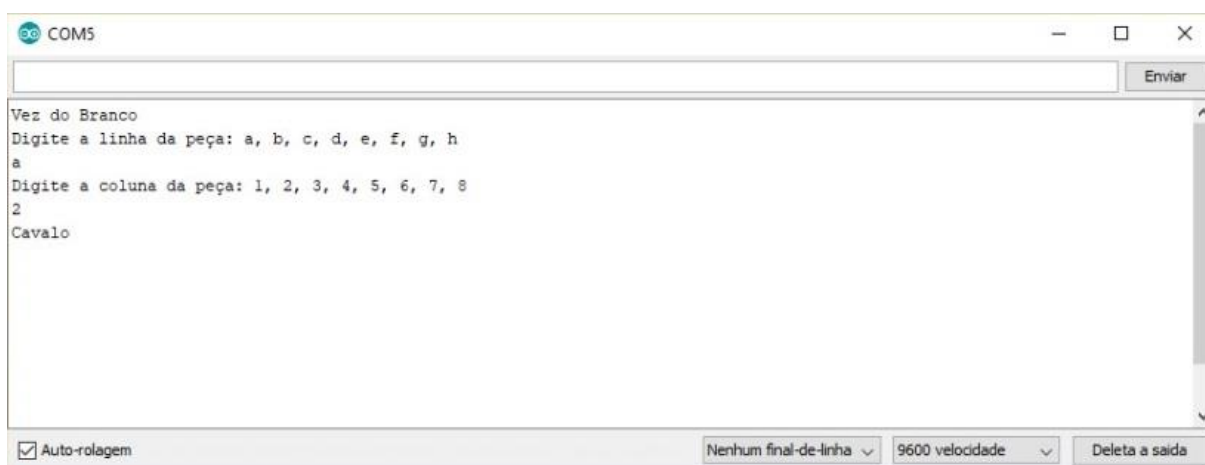
A linguagem de programação do Arduino é considerada de alto nível e é baseada em C++, com algumas pequenas modificações. Dessa forma, a maioria das funcionalidades e estruturas já são amplamente conhecidas e difundidas na comunidade.

Há duas funções padrões no Arduino: *void setup()* e *void loop()*. A primeira é executada apenas na inicialização do sistema e normalmente define parâmetros e

pinos a serem utilizadas durante o código. A segunda função é executada continuamente até o interrompimento do programa, sendo responsável pela execução propriamente dita de todo o sistema.

O tabuleiro de xadrez foi mapeado em uma matriz de inteiros, de tamanho 8x8, dentro da função *void Inicio()*, chamada pela *void setup()*. Cada peça tem sua numeração associada: Peão (1), Torre (2), Bispo (3), Cavalo (4), Rainha (5) e Rei (6). No caso de peças brancas a numeração é positiva (+) e peças pretas é negativa (-), além de posições sem peça serem zero (0).

Inicialmente o jogador seleciona através do terminal serial do Arduino, mostrado na Figura 32 uma peça a ser movida, indicando para isso a linha (letra 'a' até 'h') e uma coluna (número '1' a '8'). A função *void Leitura()* realiza a tarefa de pegar e verificar tais entradas passadas ao programa e prepará-las para as próximas etapas do código, utilizando também como auxílio as funções *void TrocaLinha()* e *void TrocaColuna()*.



**Figura 32 - Porta serial do Arduino**

Após selecionar a peça a ser movida, o programa utiliza a função *void Movimenta()* para acender as posições que a peça escolhida pode assumir. Para cada peça há uma parte individual do código que trata a sua movimentação e sua possibilidade de capturar peças adversárias. Foi importante separar o código dessa forma, pois cada peça exige uma movimentação e captura distinta das demais.



Para acender um *LED* deve-se passar para a função *void AcendeLed (int linha, int coluna)* a linha e coluna que se deseja acender. Essa função chama outras duas funções, *void SelectDemuxLinha (int s)* e *void SelectDemuxColuna (int s)*, que recebem a linha e coluna, respectivamente, através de um inteiro que varia de 0 a 7. Assim as duas funções selecionam os *selects* do respectivo *DEMUX* convertendo este número inteiro em 3 bits, acendendo tal posição.

Quando o jogador decidir para qual casa ele irá mover, o mesmo deve inserir no serial do Arduino a linha e a coluna da posição desejada, da mesma forma que o fez quando selecionou a peça a ser movida. Assim como a função *void Leitura()*, quem realiza tais ações neste caso é a função *void Leitura2()*.

Após um jogador realizar a sua vez de jogar, o programa realiza as mesmas operações para o adversário, até que um dos dois vença o jogo. Toda a parte de comunicação com os jogadores é realizada através do terminal serial do Arduino, sendo que o programa trata eventos não corretos que o jogador possa realizar, como: capturar a própria peça, tentar inserir uma posição inválida para a movimentação ou ainda tentar selecionar uma peça para movimentá-la sendo que a posição não contém nenhuma peça.

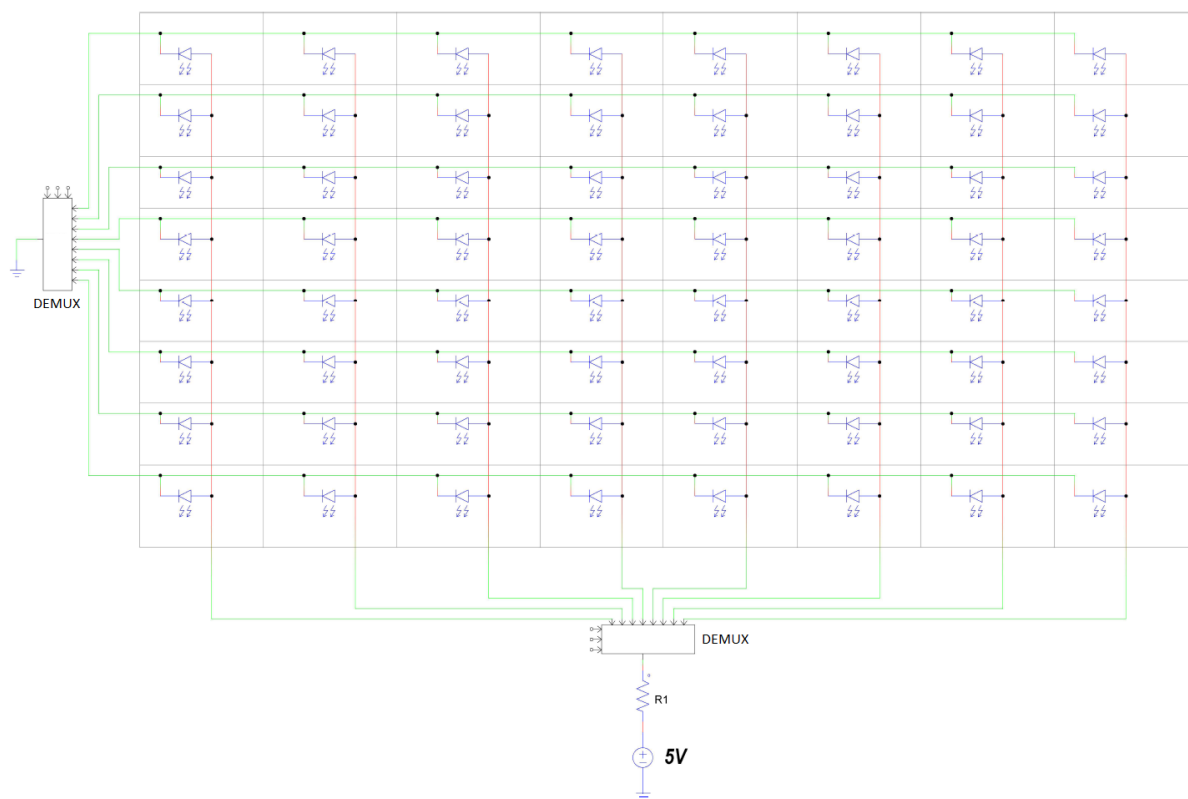
## 4.3 Eletrônica

Utilizando a interface que seleciona a peça a ser movida, fica necessário apenas criar um circuito eletrônico que acenda os *LEDs* desejados (circuito de escrita). Porém, como tentativa de deixar o jogo mais automático e direto, ou seja, sem a necessidade de um computador executando tal interface com o usuário, foi trabalhado outro circuito eletrônico que identificasse as peças do tabuleiro a cada jogada (circuito de leitura).

### 4.3.1 Circuito de escrita

Para acender um *LED* basta alimentar a sua extremidade positiva (ânodo) com 3,3V e sua extremidade negativa (cátodo) com 0V. A fim de utilizar apenas as alimentações de 5V do Arduino, se faz necessário colocar uma resistência entre a alimentação do Arduino e o ânodo do *LED*, sendo o valor de 330Ω escolhido para este

caso. Para acender os 64 *LEDs* foram utilizados dois *DEMUX TC4053BP*, um jogando 5V em uma das 8 saídas e outro jogando 0V também em uma das saídas, como mostrado na Figura 33.



**Figura 33 - Representação do circuito de escrita**

Para a elaboração do circuito físico, o circuito de escrita também foi dividido em 2 sub-circuitos com 1 *DEMUX* cada, sendo diferenciados apenas pelo valor de tensão que eles jogam em suas saídas. Foram elaborados os *PCBs* apresentados na Figura 34 e Figura 35 no *software* Proteus [12].

Selecionando os *selects* dos *DEMUXs* é possível escolher qual linha e qual coluna, ou seja, uma posição específica, para acender o *LED*. Exemplificando uma situação, quando o *DEMUX* lança 5V na coluna 1, o outro *DEMUX* lança 0V em qualquer uma das linhas, por exemplo na linha 3, assim o *LED* dessa posição recebe 0V na sua perna negativa e 3,3V na sua perna positiva, ascendendo.

Através do código, após a peça ser retirada do contato, o programa comandava os dois *DEMUX* para que eles fizessem um cruzamento de malhas e conseguissem

acionar os *LEDs* certos, mostrando as possíveis jogadas daquela peça. Porém, como o circuito pode apenas acender um *LED* por vez, foi necessário pensar numa estratégia consistente que fizesse parecer que mais de um *LED* estava aceso ao mesmo tempo.

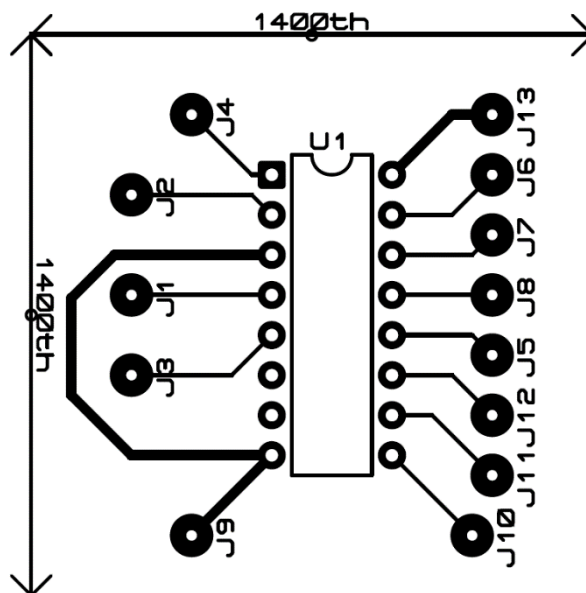


Figura 34 - Sub-circuito com *DEMUX* para o cátodo do circuito de escrita

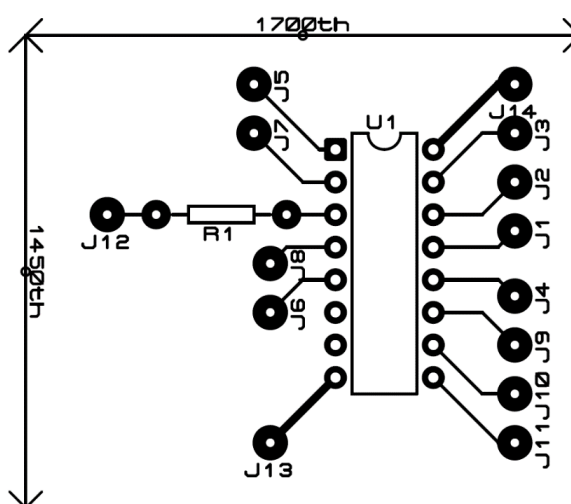


Figura 35 - Sub-circuito com *DEMUX* para o ânodo do circuito de escrita

O olho humano consegue identificar diferenças em vídeos abaixo de 30 quadros/segundo aproximadamente. Assim, se for projetado para um mesmo *LED* piscar acima dessa frequência, aparentemente ele ficaria aceso. Logo, se os *DEMUX* selecionassem o *LED* 1 (primeira coluna e primeira linha) e o acendesse, percorresse

todas as outras 63 posições rapidamente e acendesse o primeiro *LED* de novo, acima da frequência que o olho humano consegue perceber essa diferença, aparentemente não seria notado que ele piscou. Essa foi a grande ideia – simples, porém poderosa – para contornar o problema de deixar duas ou mais posições acessas ao mesmo tempo no tabuleiro.

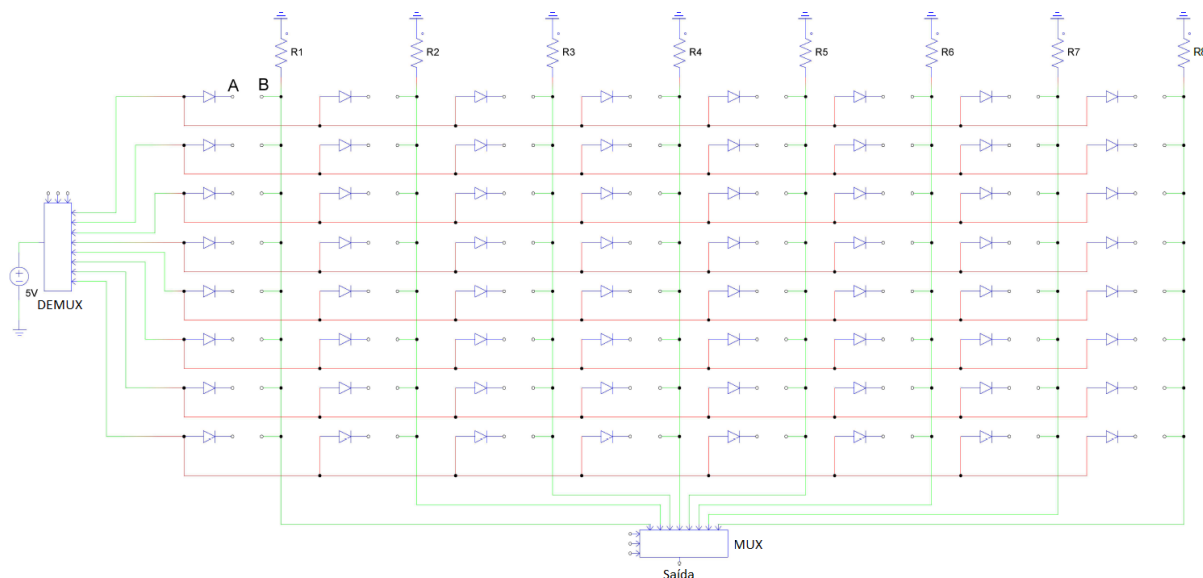
Dessa forma, todas as 3 partes essenciais do projeto (montagem mecânica, programação e circuito de escrita) foram descritas individualmente e unidas para fazer o protótipo atender ao propósito que foi construído. Até aqui é necessário que os usuários utilizem a porta serial do Arduino através de um computador com a plataforma Arduino para inserir as posições, logo deixa o jogo não dinâmico e acaba necessitando de alguém que saiba utilizar tal funcionalidade do Arduino. Dessa forma tentou-se elaborar um circuito de extensão desse projeto que eliminasse o computador para interfaceamento com o jogador e deixasse a identificação das peças totalmente automática, sendo apresentado na seção 4.3.2.

#### **4.3.2 Circuito de leitura**

O grande desafio para o circuito de leitura foi pensar em como o *software* iria identificar a presença de peça numa determinada posição. Cogitou-se a utilização de sensores indutivos, piezoelétricos e *reed switches* próximos às peças, porém qualquer uma dessas soluções iria encarecer consideravelmente o projeto.

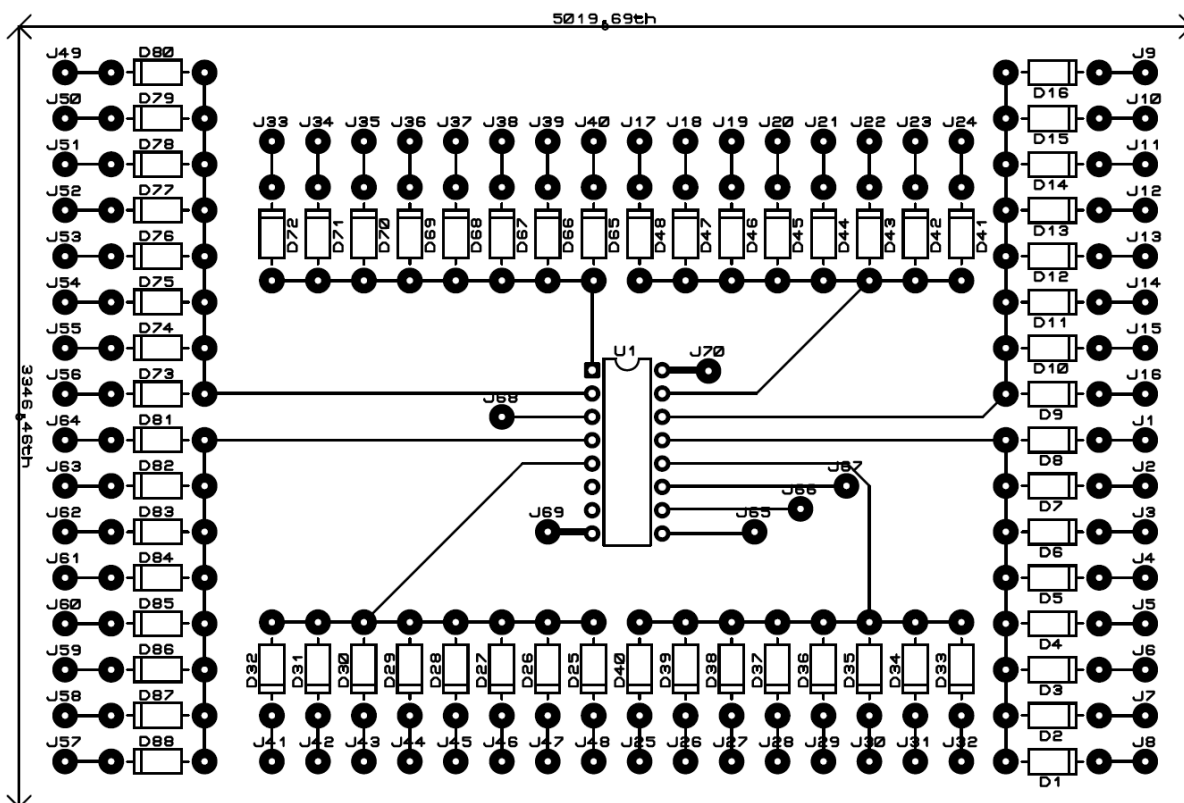
Uma última solução, mais barata, porém não tão eficiente, seria desenvolver uma espécie de chave metálica que fechasse o circuito quando a peça fosse colocada na posição em análise. O esquemático desse circuito está na Figura 36. Nele, dois fios (A e B) foram manipulados em cada superfície de encaixe no tabuleiro e duas arruelas metálicas foram coladas na base de cada peça a fim de fechar contato com os dois fios citados.

Nesse circuito, foi utilizado um *DEMUX TC4053BP* e um *MUX 74HCT251* para mapear as 64 posições do tabuleiro. Um diodo *1N4007* foi colocado em cada posição para evitar retorno de tensão e um resistor de  $330\Omega$  para cada coluna para definir 0V às colunas não selecionadas pelo *MUX* naquele momento.



**Figura 36 - Representação do circuito de leitura**

Para a elaboração do circuito físico, o circuito de leitura foi dividido em 2 sub-circuitos, um com o *MUX* e outro com o *DEMUX*, e elaborados os *PCBs* apresentados na Figura 37 e Figura 38 no *software* Proteus.



**Figura 37 - Sub-circuito com *DEMUX* para o circuito de leitura**

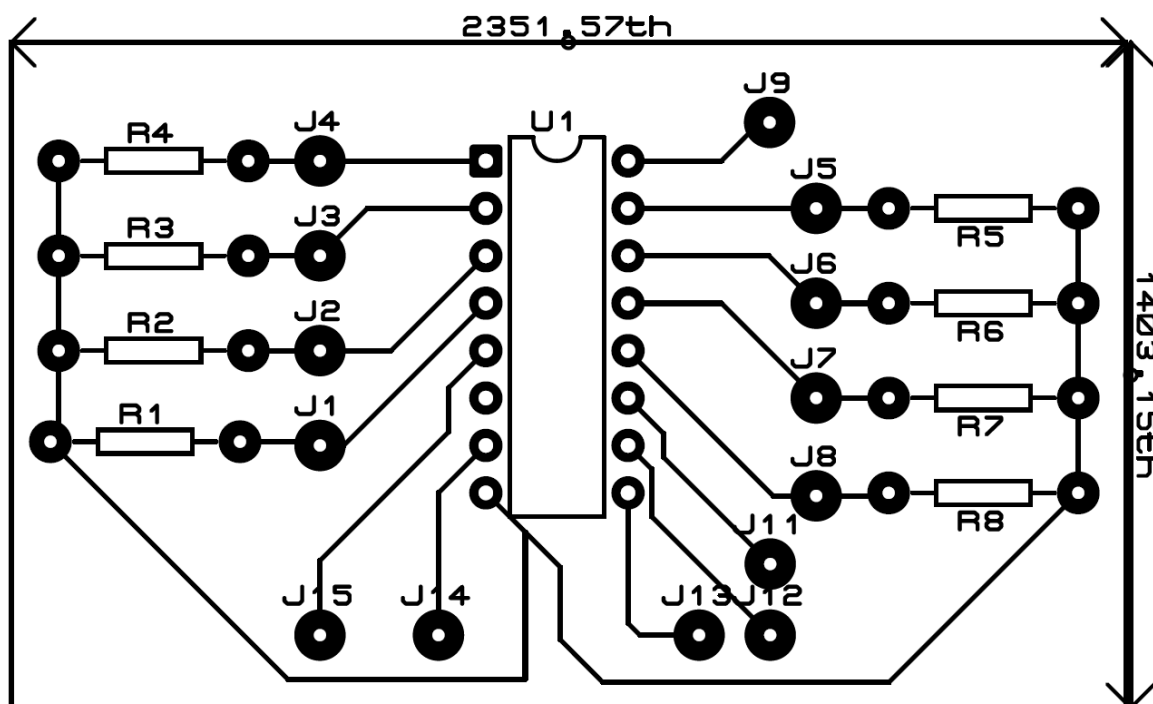


Figura 38 - Sub-circuito com *MUX* para o circuito de leitura

A lógica da solução de leitura foi a seguinte: o programa aciona os *selects* do *DEMUX*, que joga 5V na saída selecionada. Cada porta de saída do *DEMUX* foi conectada em uma linha do tabuleiro de xadrez, alimentando todas as colunas daquela linha ao mesmo tempo. O *MUX*, por sua vez, varia seus *selects* fechando contato internamente para a entrada selecionada e abrindo todas as outras 7 entradas. Dessa forma pode ser analisado a tensão na saída do *MUX*, sendo que, se houver peça naquela posição, a tensão de saída será 5V, caso contrário os contatos estarão abertos (não há peça) e a saída será 0V.

#### 4.4 Resultados da Montagem Física Completa

A interface porta serial do Arduino se mostrou muito intuitiva para o jogador e simples de ser utilizada na programação do código. Nela é permitido inserir valores de linha e coluna para realizar a inserção da posição da peça a ser movida e o destino dessa mesma peça. Além disso, a interface mostra ao usuário mensagens de erros caso este tente qualquer jogada inválida.

A solução pensada para o “circuito de escrita” acender mais de um *LED* ao mesmo tempo mostrou-se acertada e robusta. O ponto chave para a resolução desse objetivo foi, através da programação, executar a ação de acender um *LED* em alta frequência. Assim, foi possível resolver a limitação do conjunto *software/circuito* elétrico e tornar visualmente possível mostrar as posições no tabuleiro “acendendo mais de um *LED* por vez”.

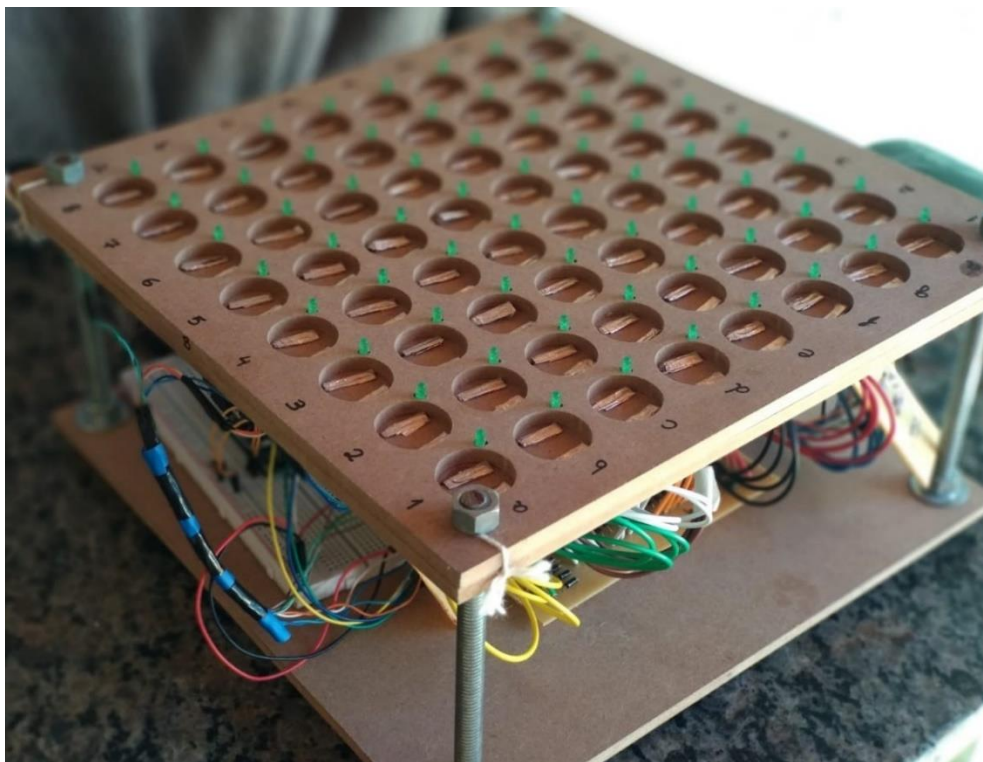
O desenvolvimento do protótipo teve início com alguns testes de bancada e utilizando uma *protoboard*, ao longo do projeto fez-se necessário a utilização de duas placas de madeira para representar o tabuleiro de xadrez. Essa solução encontrada se justifica devido a necessidade de as chaves metálicas ficarem em um nível diferente dos *LEDs*. Foi possível realizar furos na placa superior para que os mesmos abrigassem as peças com arruelas na base para que o contato com as chaves fosse o mais preciso possível.

Ao utilizar duas placas na parte superior facilitou-se também o corte delas feito em uma *CNC*, a fixação dos fios e cabos utilizados para acionamento dos *LEDs* e da identificação das peças pelo “circuito de leitura”.

Para a sustentação do tabuleiro, a alocação das ligações (cabos elétricos) e os circuitos impressos, decidiu-se furar as duas placas superiores (superfície 1 e 2), utilizar barras roscadas, arruelas e uma base de sustentação. Essa foi uma solução encontrada para facilitar possíveis alterações ao longo do projeto e uma maneira de manter organizada todas as ligações.

O resultado da montagem do protótipo, com suas respectivas visões, assim como um exemplo de peça utilizada, pode ser observado nas figuras a baixo.





**Figura 39 - Isométrico da montagem física do tabuleiro**



**Figura 40 - Visão superior da montagem física do tabuleiro**





**Figura 41 - Visão lateral da montagem física do tabuleiro**



**Figura 42 - Isométrico do peão físico**

## 5. CONCLUSÃO

Para realizar a construção de um protótipo com a finalidade pelo qual foi proposto, é necessário resolver diversos pequenos problemas de *hardware* e *software* que surgem ao longo do projeto. A ponte de união entre essas áreas ocorre com o auxílio de um microcontrolador.

A integração dessas áreas, utilizando um microcontrolador, para construção do protótipo é um grande desafio que mostra uma das aplicações, em pequena escala, de um dos possíveis papéis que o bacharel em ciência e tecnologia pode assumir no mercado de trabalho.

O protótipo criado funcionou corretamente, dentro de suas limitações de não ainda não identificar de forma totalmente automática as peças. O maior desafio do projeto como um todo foi a tentativa de implementação do “circuito de leitura”, que é uma extensão do projeto, pois a maneira como foi idealizado (utilizando uma chave metálica) não se mostrou robusto o suficiente para a resolução da proposta para o qual foi concebido, apesar de ser a solução mais barata encontrada pelo grupo.

Uma das possíveis soluções para esse circuito foi apresentada pelo orientador do projeto e consistiria em utilizar chaves fim de curso minúsculas, que normalmente são encontradas em leitores de CD/DVD. Com as características com que são fabricadas, se fossem substituídos os nossos contatos metálicos implementados por elas, seria conseguida a robustez necessária para a resolução do problema. A solução proposta aqui para tal problema não foi implementada devido ao tempo hábil de apresentação do projeto e dos gastos extras com o mesmo que não estavam viáveis no momento.

Esse projeto como um todo integrou diversas áreas do conhecimento. O protótipo construído, acaba sendo uma ótima maneira de introduzir o bacharel em um universo prático e funcional, agregando assim um grande valor teórico, técnico e científico.

## 6. REFERÊNCIAS

1. Agência Brasileira de Desenvolvimento Industrial (ABDI). **Panorama Setorial de Eletrônica para Automação.** Disponível em: <<http://www.abdi.com.br/Estudo/eletronica%20para%20automa%C3%A7%C3%A3o.pdf>>. Acesso em: 11 de Abril de 2018.
2. Wikipédia. **Automação.** Disponível em: <<https://pt.wikipedia.org/wiki/Automa%C3%A7%C3%A3o>>. Acesso em: 11 de Abril de 2018.
3. SMIDT, A.C.G. **Implementação de uma Plataforma Robótica controlada remotamente utilizando o Arduino.** São Carlos, 2013. 11 p. Disponível em: <<http://www.tcc.sc.usp.br/tce/disponiveis/97/970010/tce-19112013-093717/?&lang=br>>. Acesso em: 11 de Abril de 2018.
4. OP.CIT. **Tradução de Lauand.** [S.l.]: [s.n.], 49-51 p.
5. Xadrez Center. **O xadrez e a tomada de decisões.** Disponível em: <<http://xadrezcentroaweaw.blogspot.com/2011/07/tcc-o-xadrez-e-tomada-de-decisoes.html>>. Acesso em: 11 de Abril de 2018.
6. Confederação Brasileira de Xadrez (CBX). **Lei do Xadrez da FIDE.** Tradução de Antonio Bento. Revisão de Igor Lutz. [S.l.]: [s.n.]. Disponível em: <[http://www.cbx.org.br/files/downloads/Xadrez\\_lei\\_da\\_FIDE.pdf](http://www.cbx.org.br/files/downloads/Xadrez_lei_da_FIDE.pdf)>. Acesso em: 11 de Abril de 2018.
7. Arduino. **Download.** Disponível em: <<https://www.arduino.cc/en/Main/Software>>. Acesso em 09 de Fevereiro de 2018.
8. Embarcados. **Arduino Leonardo.** Disponível em: <<https://www.embarcados.com.br/arduino-leonardo/>>. Acesso em: 11 de Abril de 2018.

- 
9. Dassault Systemes. **SolidWorks**. Disponível em: <<https://www.solidworks.com/sw/support/downloads.htm>>. Acesso em: 04 de Maio de 2017.
10. GrabCAD. **Steeline Chess Set Concept**. Disponível em: <<https://grabcad.com/library/steeline-chess-set-concept-1>>. Acesso em: 04 de Maio de 2017.
11. GitHub. **TCIC**. Disponível em: <<https://github.com/JTOLEDOJ/Tabuleiro-de-Xadrez-Interativo>>. Acesso em: 24 de Junho de 2018.
12. Labcenter. **Proteus**. Disponível em: <<https://www.labcenter.com/downloads/>>. Acesso em: 23 de Agosto de 2017.