



ETEC JORGE STREET

TRABALHO DE CONCLUSÃO DO CURSO TÉCNICO EM MECATRÔNICA

TIC TAC TOE

**Eric Rinaldi
Felipe Pinheiro Torres
Henry Fugolim Lima
Luidgi Augusto Donadio Campos
Pedro Assunção Duó
Thiago Sales de Oliveira
Thiago Yuiti Tomaru
Victor Vicente Casimiro**

**Professores Orientadores:
Larry Aparecido Aniceto
Luiz Antônio Carnielli**

**São Caetano do Sul / SP
2019**

TIC TAC TOE

Trabalho de Conclusão de Curso
apresentado como pré-requisito para
obtenção do Diploma de Técnico em
Mecatrônica.

**São Caetano do Sul / SP
2019**

AGRADECIMENTOS

Às nossas famílias que nos incentivaram a concluir o curso com patrocínio financeiro aos nossos projetos e, principalmente, apoio moral.

Aos Profs. Larry e Carnielli que alicerçaram o ensino e a aprendizagem a fim de que este projeto pudesse ser concretizado, com apoio e esclarecimentos.

A Edson Militão, membro da equipe de manutenção da Etec Jorge Street, que foi fundamental no auxílio à produção da parte mecânica do projeto.

Ao Prof. William que alicerçou no desenvolvimento da lógica do projeto, em especial na programação do Arduino.

A todos os Professores que participaram da construção do conhecimento dos alunos, pois sem esse auxílio, possivelmente este projeto não teria se transformado em realidade e edificado um sonho.

Ao prof. Filliputti que auxiliou com a corrosão e com o circuito da placa.

Aos colegas do curso com os quais tivemos oportunidade de conviver durante a aquisição da aprendizagem e repartir incertezas na caminhada em busca desta ascensão cultural. Em específico a equipe de César de Moura que puderam nos ajudar diretamente com nossas dificuldades.

RESUMO

O projeto tem como intuito principal entreter os jovens que vivem na era da modernização e robótica em um jogo clássico que é o da velha. Pois após uma breve observação chegou-se à conclusão de que cada vez menos estudantes socializam, como antigamente, e passam a maior parte do tempo em celulares e tablets. Para isso foi utilizado uma ponte rolante, um tabuleiro inteligente e Arduino com logicas de programação para que tudo funcione conforme o necessitado. O projeto acabou englobando todos os componentes curriculares de mecatrônica (eletrônica, automação, mecânica e elétrica), por isso deduziu-se que ele seria ideal para o projeto de conclusão do curso.

Palavras-chave: socialização; tecnologia; entretenimento.

The project's main purpose is to entertain young people living in the age of modernization and robotics in a classic game which one is the "tic tac toe". After a brief observation, it has been concluded that a great number of students do not socialize, as in old times, and spend most of their time on mobile phones and tablets. Considering all of it, we used an overhead crane, a smart pegboard and an Arduino with programming logics so that everything works as needed. The project ended up encompassing all mechatronics curriculum components (electronics, automation, mechanical and electrical), so it has been assumed it would be ideal for the course completion project.

Keywords: socialization; technology; entertainment.

LISTA DE FIGURAS

Figura 1 – Exemplo de ponte rolante	11
Figura 2 – Exemplo de funcionamento do sensor óptico.....	13
Figura 3 – Sensor utilizado no tabuleiro Tic Tac toe	13
Figura 4 – Motor Mabuchi.....	14
Figura 5 – Placa de Arduino Mega	14
Figura 6 – Circuito elétrico do painel e dos motores	15
Figura 7 – Exemplo de ponte rolante	15
Figura 8 – Chaves de turno	16
Figura 9 – Cabos de presença no circuito	16
Figura 10 – Sensores ópticos no circuito.....	17
Figura 11 – LEDs de turno no circuito	17
Figura 12 – LEDs de localização das peças no circuito	18
Figura 13 – Relés do painel e relé geral.....	18
Figura 14 – Circuito elétrico do painel e interface de potência	19
Figura 15 – Diagrama em blocos do processo	21
Figura 16 – Fluxograma do processo	22
Figura 17 – Esboço do carro menor	23
Figura 18 – Esboço da peça.....	24
Figura 19 – Desenho da estrutura de aço	24
Figura 20 – Desenho do carro menor.....	25
Figura 21 – Desenho do carro maior	25
Figura 22 – Custo Hora-Homem	26
Figura 23 – Estrutura Tic Tac Toe recém finalizada.....	27
Figura 24 – instrumento de solda utilizado na parte mecânica.....	27
Figura 25 – Acabamento das barras da estrutura com esmeril	28
Figura 26 – Roda do carro menor	28
Figura 27 – Carros Prontos	29
Figura 28 – Vista dos carros na estrutura.....	29
Figura 29 – Estado do carro maior na estrutura final.....	30
Figura 30 – Peça Torneada.....	30
Figura 31 – Esboço da peça.....	30
Figura 32 – Teste do carro em movimento.....	31

Figura 33 – Cortes do painel	31
Figura 34 – Painel cortado	32
Figura 35 – Painel pronto	32
Figura 36 – Montagem do painel com parafusos	33
Figura 37 – Painéis conectados à estrutura	33
Figura 38 – Tabuleiro com os cantos de encaixe em destaque	34
Figura 39 – Estrutura com o tabuleiro destacado	34
Figura 40 – Placa com circuito em caneta.....	35
Figura 41 – Placa submersa em água.....	35
Figura 42 – Placa submersa em percloroeto de ferro e água.....	36
Figura 43 – Localização da solda do sensor na placa	36
Figura 44 – Localização da solda dos dois resistores	37
Figura 45 – Trilhas dos sinais de entrada entre placa e Arduino.....	37
Figura 46 – Cabos de entrada soldados na placa	37
Figura 47 – Relés geral e de painel.....	38
Figura 48 – Solda dos cabos nos LEDs	38
Figura 49 – Solda dos cabos no joystick	39
Figura 50 – Solda dos cabos nos botões e chaves	39
Figura 51 – Interface de Potência	39
Figura 52 – Teste com protoboard	40
Figura 53 – Desenvolvimento da programação	40
Figura 54 – Fluxograma e matriz 3x3	41
Figura 55 – Fluxograma e exemplo de vitória	41
Figura 56 – Peças do Tic Tac Toe	63
Figura 57 – Tic Tac Toe	64

Sumário

Introdução	8
1 – Fundamentação Teórica.....	11
1.1 – Pontes Rolantes	11
1.2 – Sensores Ópticos	12
1.2.1 – Sensor óptico difuso.....	12
1.3 – Motores Mabuchi	13
1.4 – Arduino Mega	14
2 – Planejamento do Projeto	15
2.1 – Parte elétrica/eletrônica.....	15
2.2 – Diagrama em blocos.....	21
2.3 – Parte lógica	22
2.4 – Parte Mecânica.....	23
2.5 – Cronograma Geral e Cargos	26
3 – Desenvolvimento do Projeto.....	27
3.1 – Desenvolvimento do Projeto.....	27
3.2 – Desenvolvimento dos Carros Móveis	28
3.3 – Desenvolvimento dos Painéis	27
3.4 – Desenvolvimento do Tabuleiro	33
3.5 – Desenvolvimento da parte elétrica/eletrônica.....	34
3.6 – Desenvolvimento da Programação.....	39
3.7 – Produção das peças.....	62
4 – Resultados Obtidos	64
Conclusão	66
Referências	67

Introdução

O que é o TIC TAC TOE?

O TIC TAC TOE é um meio para prática de qualquer jogo de tabuleiro. Consiste em uma ponte rolante que trabalha nos três eixos (X, Y e Z) com dois carros (X e Y) e um sistema de içamento (Z), para isso foi empregado o uso de três motores, que atuam de forma isolada ou simultânea.

Já o tabuleiro – feito de acrílico – conta com 9 sensores óticos (o protótipo foi feito baseado no jogo da velha, que tem o tabuleiro 3x3) capazes de identificar qual peça está em cada casa, com isso pode-se acusar quando um jogador ganhar ou der empate.

Enquanto o circuito que comanda tudo é em sua maior parte digital, com o uso do Arduino para o acionamento da ponte e do tabuleiro foi possível otimizar o espaço que o projeto ocupa, simplificar sua produção e facilitar a manutenção, podendo inclusive, mudar a lógica a qualquer momento, alterando assim o jogo.

O Arduino foi criado em 2005 por um grupo de 5 pesquisadores: Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis. O objetivo era elaborar um dispositivo que fosse ao mesmo tempo barato, funcional e fácil de programar, sendo dessa forma acessível a estudantes e projetistas amadores. Além disso, foi adotado o conceito de hardware livre, o que significa que qualquer um pode montar, modificar, melhorar e personalizar o Arduino, partindo do mesmo hardware básico. [...]

Ref.: (disponível em: <https://www.filipeflop.com/blog/o-que-e-arduino/>. Acesso em setembro de 2019.)

Tema e delimitação.

CURSO: Mecatrônica

TEMA: Automação, Eletrônica Analógica e Digital e Tecnologia Mecânica.

Delimitação do tema: Jogo da velha automatizado.

Objetivos – geral e específicos

O avanço tecnológico trouxe diversas dádivas, mas como tudo no mundo, também tem seus pontos negativos: o isolamento social do jovem. As crianças e adolescentes nos dias de hoje preferem muito mais a interação através de uma tela do que pessoalmente, e não ter uma forma de fundir a

tecnologia com a interação presencial entre humanos tem incomodado cada vez mais nos últimos anos.

O Brasil já conta com 22 milhões dos chamados nativos digitais, nascidos e criados a partir da década 1980, na era dos games e da internet. Contrariando prognósticos de que a tecnologia apenas ajudaria a multiplicar informações e o círculo de amizades, muitas crianças e adolescentes nunca estiveram tão desconectados do mundo. Parecem hipnotizados por seus aparelhos móveis, perdendo a vontade de estudar, de brincar ao ar livre e até de conversar entre si e com os familiares, sem intermediação das telas. É o que mostra a partir de hoje série de reportagens do EM sobre o problema, que tem levado especialistas a alertar pais para o abuso e a falta de controle dos adeptos. [...]"

Ref.:(Jorge Macedo - especial para o EM MINAS GERAIS, Brasil, 2014. Disponível em: https://www.em.com.br/app/noticia/gerais/2014/05/25/interna_gerais,532336/exagero-de-tecnologia-deixa-criancas-e-adolescentes-desconectados-do-mundo-real.shtml . Acesso em setembro de 2019.)

O projeto foi estruturado em uma ponte rolante, pois é uma tecnologia usada em jogos de fliperamas para garantir maior velocidade em guias de bichos de pelúcia. Nós levamos a diversão desta tecnologia para o jogo da velha, inovando-o com a ponte.

Justificativa

O projeto visa tornar mais atrativo para as crianças e adolescentes os jogos de tabuleiro, incentivando-as a interagir com outras e quem sabe, se interessar por tecnologia de automação e programação.

Metodologia

Utilizando a infraestrutura da escola disposta em oficinas mecânicas e laboratórios eletroeletrônicos, pretendemos construir uma grua.

Para isso, empregamos o uso de aço galvanizado para a confecção da estrutura, motores do tipo mabuchi que realizam movimentação nos três eixos. No que tange a parte eletroeletrônica foi utilizado Arduino mega e placa de inversão de rotação dos motores com relê.

1 – Fundamentação Teórica

Neste capítulo, serão apresentados tecnologias e inspirações que influenciaram na escolha do projeto Tic Tac Toe.

1. 1 - Pontes Rolantes

Foi utilizado no projeto Tic Tac Toe, o sistema de pontes rolantes. Aplicamos o uso das mesmas em nosso ambiente de jogo para garantir a velocidade de içamento das peças e transportá-las com facilidade ao local desejado, baseando-se nos serviços empregados em áreas industriais no transporte de cargas com pontes.

As pontes rolantes são comuns no içamento e transporte de cargas diversas. As pontes rolantes se movem sobre trilhos, apoiados em vigas de rolamento, a carga de içamento da ponte rolante é deslocada tridimensionalmente, limitada apenas pelo vão do equipamento, pelo comprimento do caminho de rolamento e pela altura de elevação. As pontes viga simples utilizam talhas acopladas sob a viga, enquanto as de viga dupla utilizam carros-guincho posicionados sobre elas. As pontes rolantes CSM dispõem de recursos tecnológicos de operação simplificada que impactam positivamente na segurança e conforto dos operadores, velocidade dos processos fabris, redução do consumo de energia elétrica e baixos níveis de ruído no ambiente de trabalho.

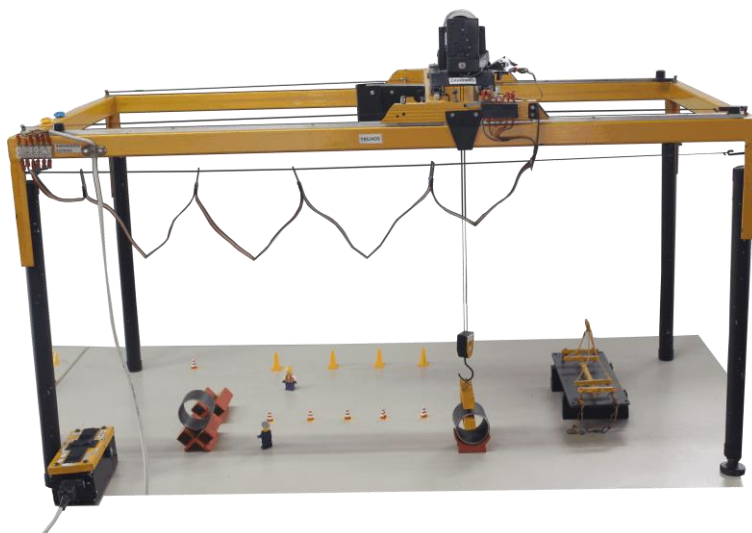


Figura 1 – Exemplo de ponte rolante.

1.2 – Sensores Ópticos

Os **sensores Ópticos** ou fotoelétricos tem como princípio de funcionamento o uso da propagação da luz, este tipo de sensor é utilizado comumente para indexação de objetos ou para medições de distância em que um objeto se encontra em relação ao sensor. A luz emitida pelos sensores óticos pode ser dos seguintes tipos:

- Vermelha;
- Laser Vermelho;
- Infravermelho.

Cada tipo de luz é indicado para uma determinada aplicação, por exemplo, a luz VERMELHA é indicada para detecção de objetos opacos de médio e grande porte, como caixas de papelão e embalagens não metalizadas. Já a luz LASER é utilizada para detecções mais precisas envolvendo objetos de pequeno porte, devido ao feixe de emissão da luz ser estreito e focalizado. Por último, a luz INFRAVERMELHA é utilizada quando há a necessidade de se detectar objetos transparentes, como vidro, garrafas plásticas entre outros objetos.

1.2.1 - Sensor óptico difuso

Foram utilizados no jogo da velha, sensores ópticos reflexivos do tipo difuso, que possuem um LED emissor de raios infravermelhos e ao mesmo tempo um transistor fotorreceptor em seu interior. O emissor emite uma faixa de luz que dependendo da intensidade que for refletida através de uma superfície no receptor polariza o transistor e causa uma determinada queda de tensão entre coletor e emissor do mesmo, possibilitando assim realizar a detecção de cores dependendo da queda de tensão entre os terminais do transistor. Conforme pesquisado no site Citisystems, os sensores difusos:

Possuem como característica principal o fato de o emissor e o receptor de luz estar no mesmo componente, dispensando o uso de outro componente com característica receptora ou de reflexão. O emissor envia um feixe de luz, podendo ser de qualquer tipo, e ao um objeto refletor passar por este feixe de luz, a luz é refletida pelo objeto de volta para a unidade receptora, que está localizada no mesmo componente. Este tipo de Sensor Óptico possui um range de detecção inferior aos outros tipos, devido à reflexão depender exclusivamente do objeto a ser analisado, lembrando que quanto mais escuro for o objeto menor será o range de detecção. Os

receptores difusos não bloqueiam luz não polarizada, permitindo que o mesmo identifique a luz refletida pelo objeto.

A figura seguinte representa o funcionamento de um sensor difuso de luz vermelha.

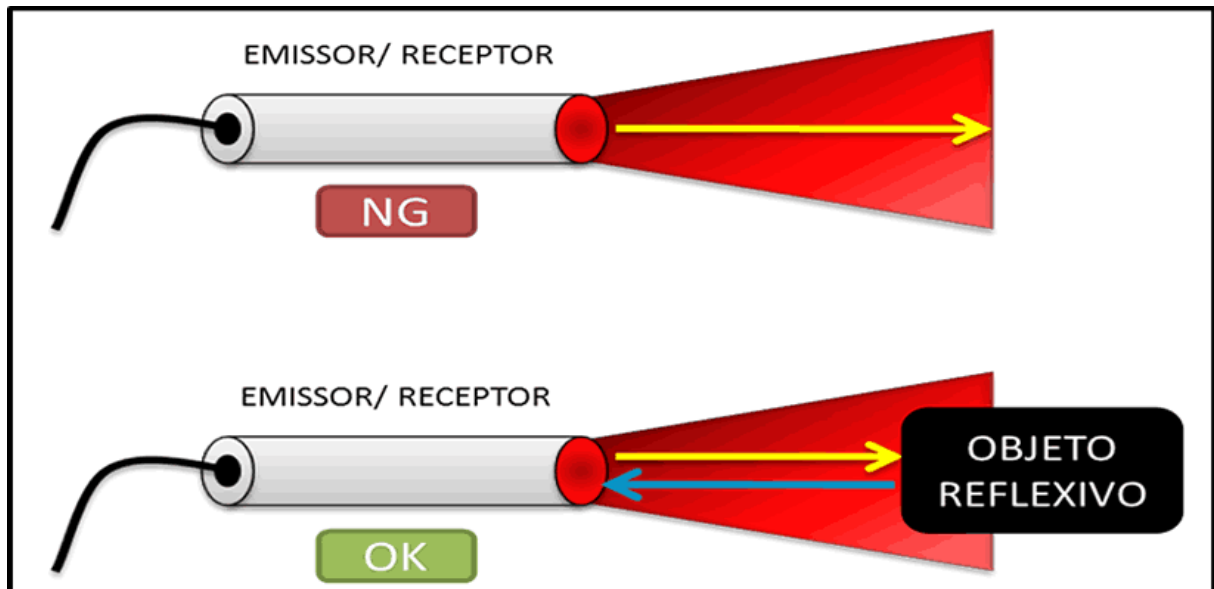


Figura 2 - Exemplo de funcionamento do sensor óptico difuso de luz vermelha



Figura 3 – Sensor utilizado no tabuleiro Tic Tac toe (modelo TCRT 5000)

1.3- Motores Mabuchi

A ponte rolante é fundamental para o deslocamento das peças, mas isso não poderia ser feito sem a integração de motores na estrutura. Para isso, usamos motores de vidro de carro da marca Mabuchi, em razão de seu ótimo custo-benefício.

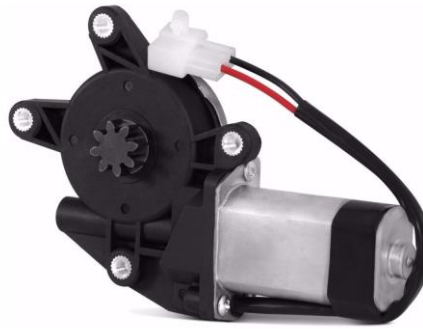


Figura 4 – motor mabuchi

1.4– Arduino Mega

Para programar o funcionamento do painel e do tabuleiro do jogo, utilizamos a placa Arduino MEGA, que possui um total de 54 portas digitais. Utilizar a placa simboliza uma grande facilidade para o projeto que utiliza 45 portas lógicas, incluindo todas as entradas e saídas. Segundo o site Filipeflop, “O Arduino Mega 2560 R3 é a placa perfeita para quem procura fazer projetos mais elaborados que precisam de um maior número de entradas e saídas[...]”. Através da alta fluidez de dados da placa e de sua grande capacidade de memória é possível manter o circuito e as atividades do projeto integrados por meio da programação da placa em linguagem C.

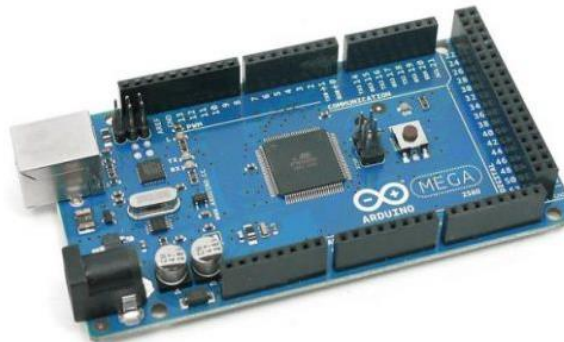


Figura 5 – Placa de Arduino Mega

2 – Planejamento do Projeto

2.1 - Parte elétrica/eletrônica:

- Circuito integrado

Este tópico mostra o circuito completo do projeto que integra toda a parte do hardware onde se incluem a placa Arduino Mega, os joysticks e os motores da ponte rolante.

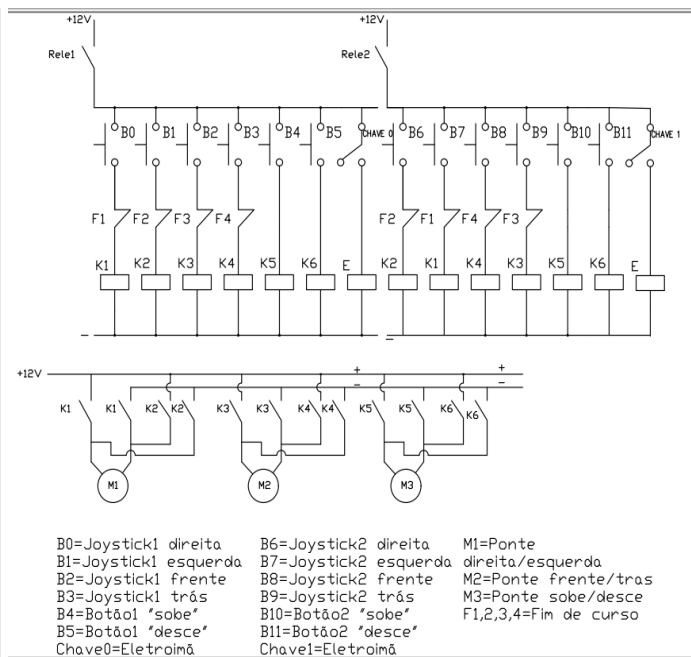
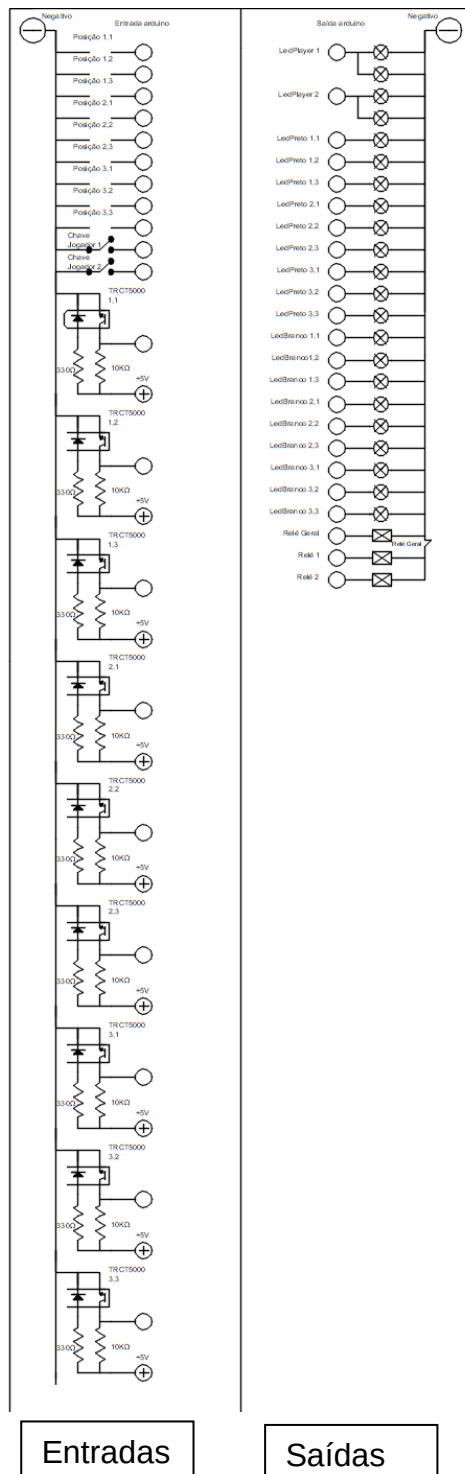


Figura 6 – Circuito elétrico do painel e dos motores

Figura 7 – Circuito elétrico do tabuleiro e Arduino

À esquerda, temos o circuito do Arduino com suas respectivas entradas e saídas.

Começando pelas chaves de jogador 1 e jogador 2, podemos definir ao iniciar a partida, qual jogador posicionará a primeira peça e em seguida passar o turno para o próximo jogador.

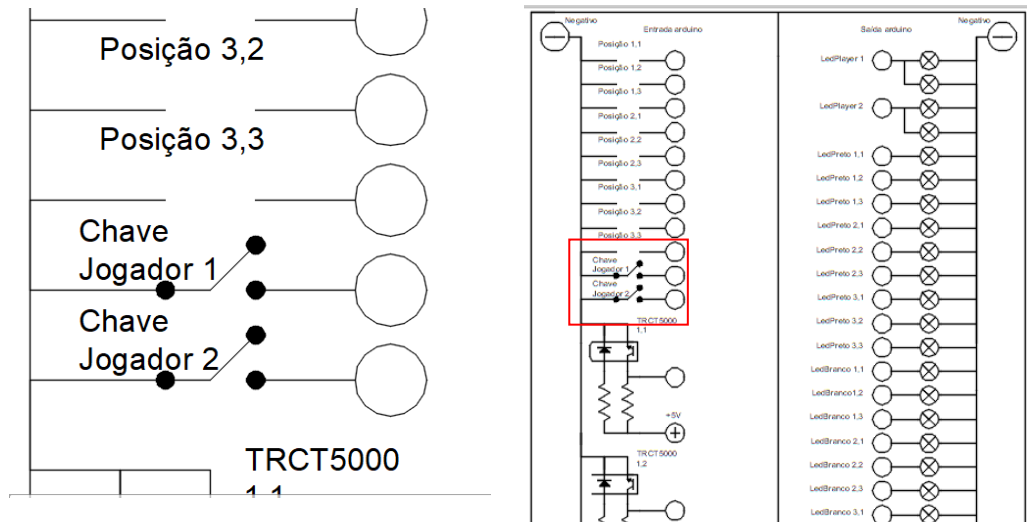


Figura 8 – Chaves de turno

Em seguida, conectados à entrada do Arduino temos 9 cabos de presença, sendo denominados desta forma devido a abertura presente em seu circuito, que posteriormente será fechada quando a peça – feita de material condutor - estiver posicionada sobre os cabos conduzindo a corrente e fechando o circuito. Feito isso, o Arduino irá identificar que existe uma peça naquela casa.

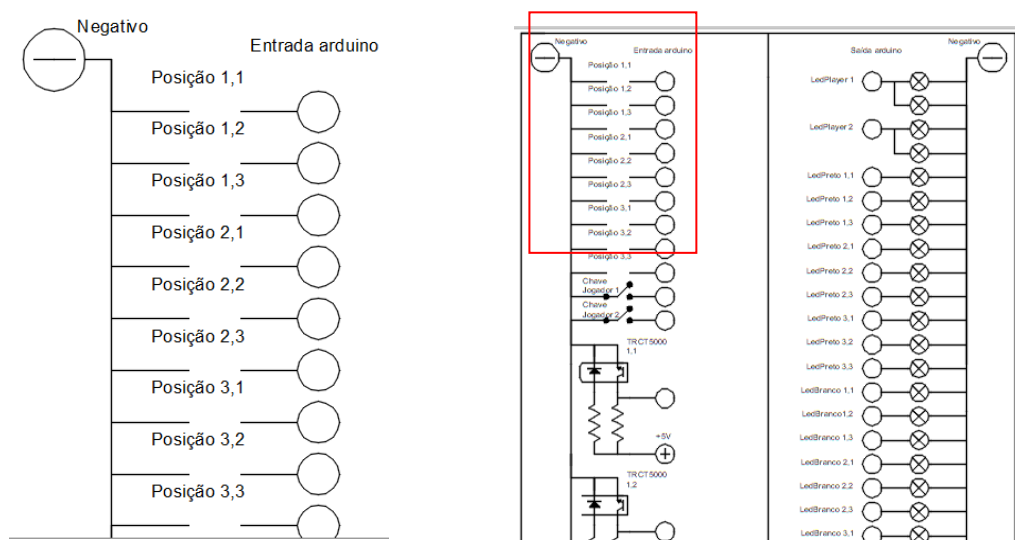


Figura 9 – Cabos de presença no circuito

Ainda nas entradas temos também 9 sensores ópticos que farão o papel de detectar se as peças são de cor preta ou branca para então passar essas informações ao Arduino. A figura a seguir mostra o sensor integrado ao circuito.

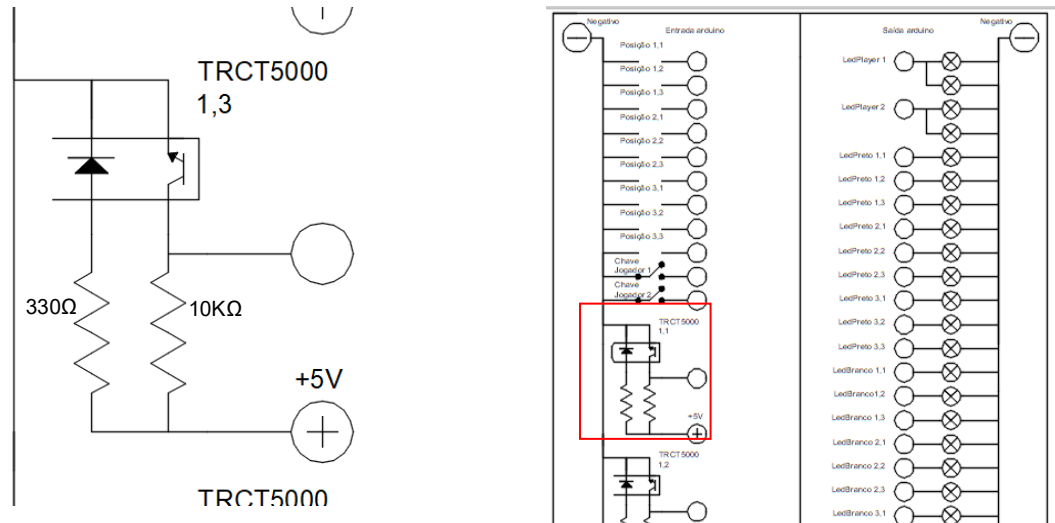


Figura 10 – Sensores ópticos no circuito

O circuito sensorial é composto por um LED infravermelho e um fototransistor, energizados por uma tensão de 5v, mas antes protegidos por duas resistências.

Nas saídas do Arduino temos 4 LEDs que indicarão para o jogador quando este deverá realizar sua jogada e quando deve esperar a vez de seu adversário. Cada painel terá 2 LEDs para as respectivas ações.

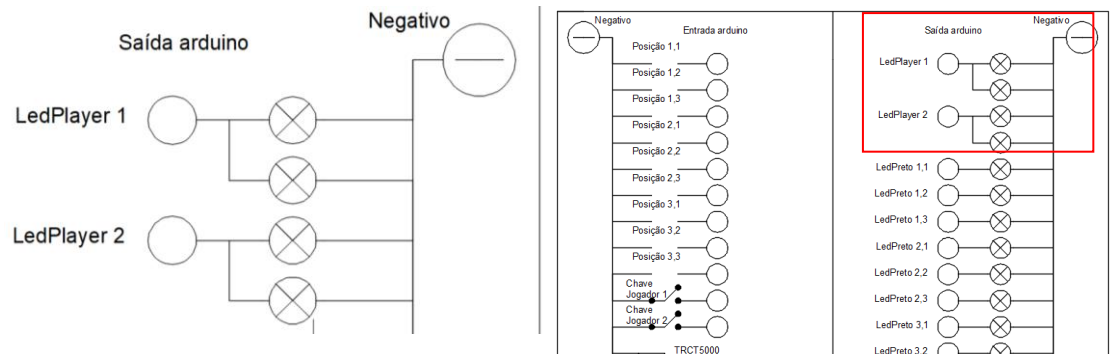
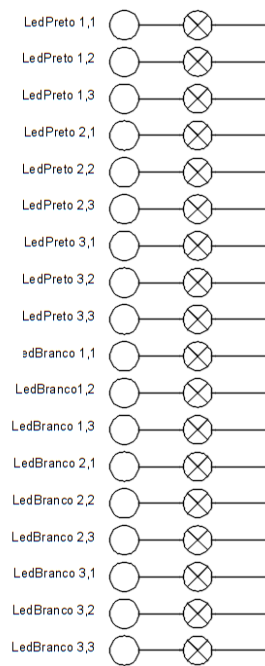


Figura 11 – LEDs de turno no circuito

As saídas também possuem 18 LEDs, 9 para cada painel de jogador podendo ser painel preto ou branco. Estes LEDs indicam ao jogador em qual casa do jogo da velha estão localizadas suas peças.

Figura 12 – LEDs de localização das peças no circuito



O jogo da velha é composto por uma matriz 3x3, portanto possui 9 casas, sendo necessários o mesmo número para os sensores, LEDs de painel e cabos de presença.

Por fim, temos o relé geral responsável por alimentar os dois painéis liberando assim o uso dos joysticks, eletroímã, motores e dos outros botões.

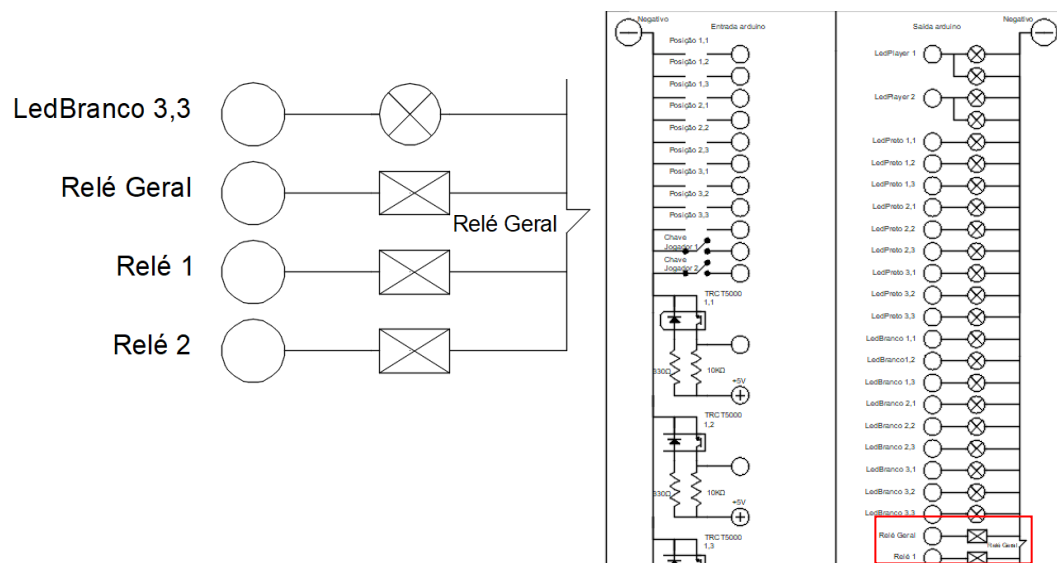


Figura 13 – Relés do painel e relé geral

Os relés 1 e 2 irão energizar o circuito elétrico de controle dos dois jogadores, com uma tensão de 12V.

No circuito dos painéis temos os joysticks onde cada um deles constitui de 4 botões que direcionam a ponte rolante em x e y. Como são 2 painéis e 2 joysticks, serão utilizados 8 botões que acionarão 4 relés para que seja realizado o movimento dos motores. No circuito elétrico do projeto, retratado abaixo (figura 14), temos que B0 até B3 são os botões que acionarão os relés K1 a K4 e que consequentemente seus contatos irão ligar os motores M1 e M2. A chave 0 aciona o eletroímã e os botões B4 e B5 acionam o motor M3 que movimenta a ponte em z. As mesmas configurações se aplicam para o circuito posterior, do jogador 2. Os contatos F1 a F4 são fins de curso utilizados para limitar o deslocamento dos motores dentro da área do projeto e garantir a segurança na estrutura.

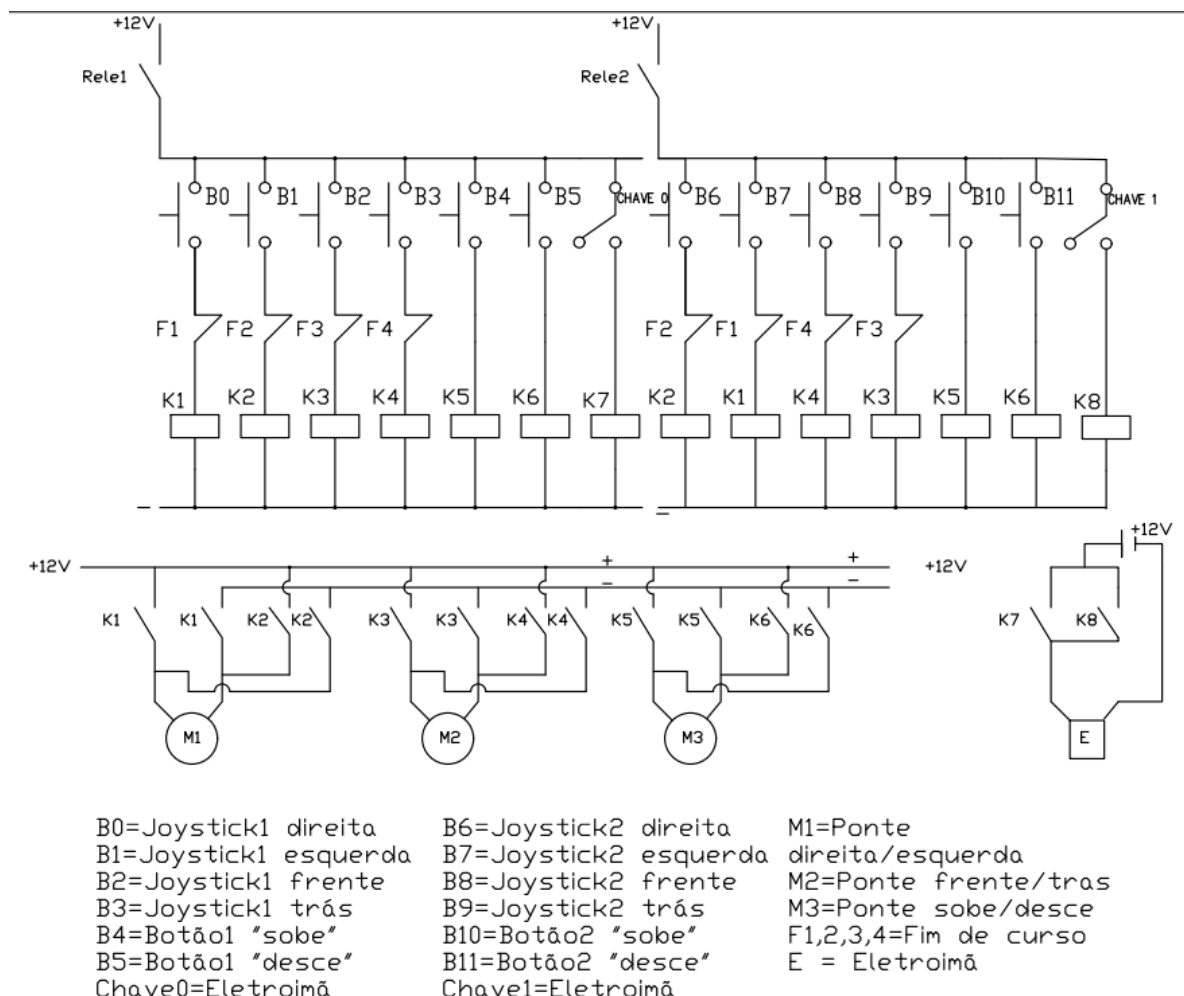


Figura 14 – Circuito elétrico do painel e interface de potência

Como mostrado na figura, o projeto integra a ligação dos motores em uma interface de potencia, com o objetivo de inverter a rotação dos motores e garantir os movimentos cima/baixo, direita/esquerda, frente/trás. Observa-se também que o eletroímã é acionado por uma fonte de 12V que objetifica uma melhor distribuição de corrente para garantir o funcionamento do mesmo.

- Entradas e Saídas

Esta parte mostra uma lista com todos os componentes usados no Tic Tac Toe, separados em entrada e saída.

➤ Parte Elétrica

Entrada	Saída
B0 = joystick P1(direita)	Motor 1(Ponte direita/esquerda)
B1 = joystick P1(esquerda)	Motor 2(Ponte frente/trás)
B2 = joystick P1 (frente)	Motor 3(Ponte sobe/desce)
B3 = joystick P1 (trás)	K1 = relé (Ponte direita/esquerda)
B4 = botão P1(sobe)	K2 = relé (Ponte direita/esquerda)
B5 = botão P2 (desce)	K3 = relé (Ponte frente/trás)
Chave 0 = chave do eletroímã P1	K4 = relé (Ponte frente/trás)
Chave 1 = chave do eletroímã P2	K5 = relé (Ponte sobe/desce)
B6 = joystick P2(direita)	K6 = relé (Ponte sobe/desce)
B7 = joystick P2(esquerda)	
B8 = joystick P2(frente)	
B9 = joystick P2(trás)	
B10 = botão P2(sobe)	
B11 = botão P2(desce)	

➤ Parte Eletrônica - Arduino

Entradas	Saídas
Cabo de presença (1,1)	LED Player 1
Cabo de presença (1,2)	LED Player 1
Cabo de presença (1,3)	LED Player 2
Cabo de presença (2,1)	LED Player 2
Cabo de presença (2,2)	LED painel preto (1,1)
Cabo de presença (2,3)	LED painel preto (1,2)

Cabo de presença (3,1)	LED painel preto (1,3)
Cabo de presença (3,2)	LED painel preto (2,1)
Cabo de presença (3,3)	LED painel preto (2,2)
Chave jogador 1	LED painel preto (2,3)
Chave jogador 2	LED painel preto (3,1)
Sensor TCRT 5000 (1,1)	LED painel preto (3,2)
Sensor TCRT 5000 (1,2)	LED painel preto (3,3)
Sensor TCRT 5000 (1,3)	LED painel branco (1,1)
Sensor TCRT 5000 (2,1)	LED painel branco (1,2)
Sensor TCRT 5000 (2,2)	LED painel branco (1,3)
Sensor TCRT 5000 (2,3)	LED painel branco (2,1)
Sensor TCRT 5000 (3,1)	LED painel branco (2,2)
Sensor TCRT 5000 (3,2)	LED painel branco (2,3)
Sensor TCRT 5000 (3,3)	LED painel branco (3,1)
	LED painel branco (3,2)
	LED painel branco (3,3)
	Relé geral
	Relé 1
	Relé 2

2.2 - Diagrama em Blocos

Para facilitar o entendimento do processo integrado do projeto, este também pode ser acompanhado pelo diagrama abaixo (figura 15).

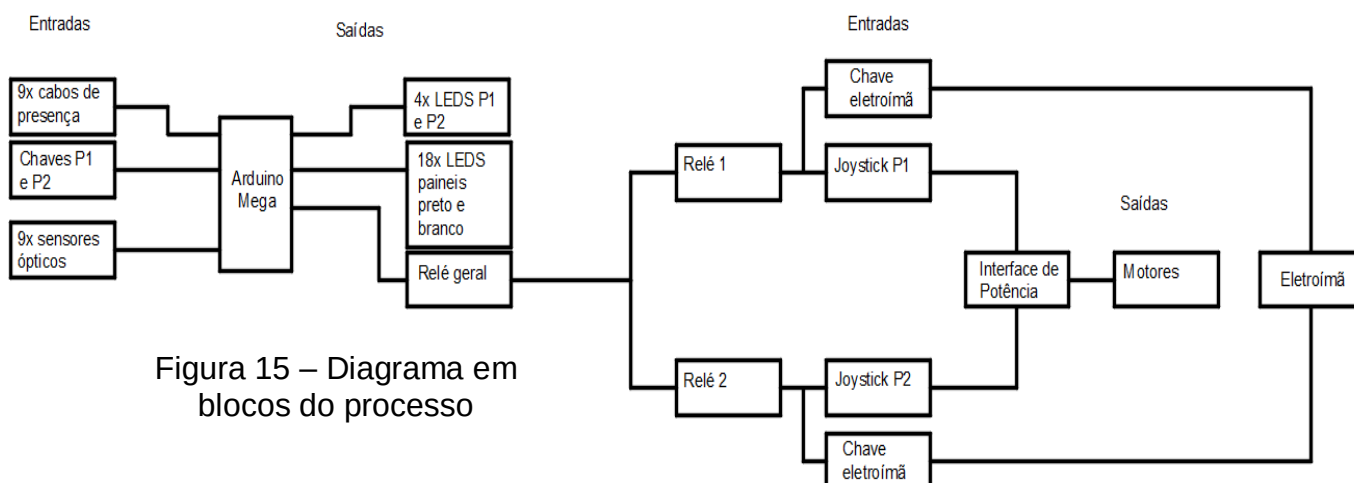
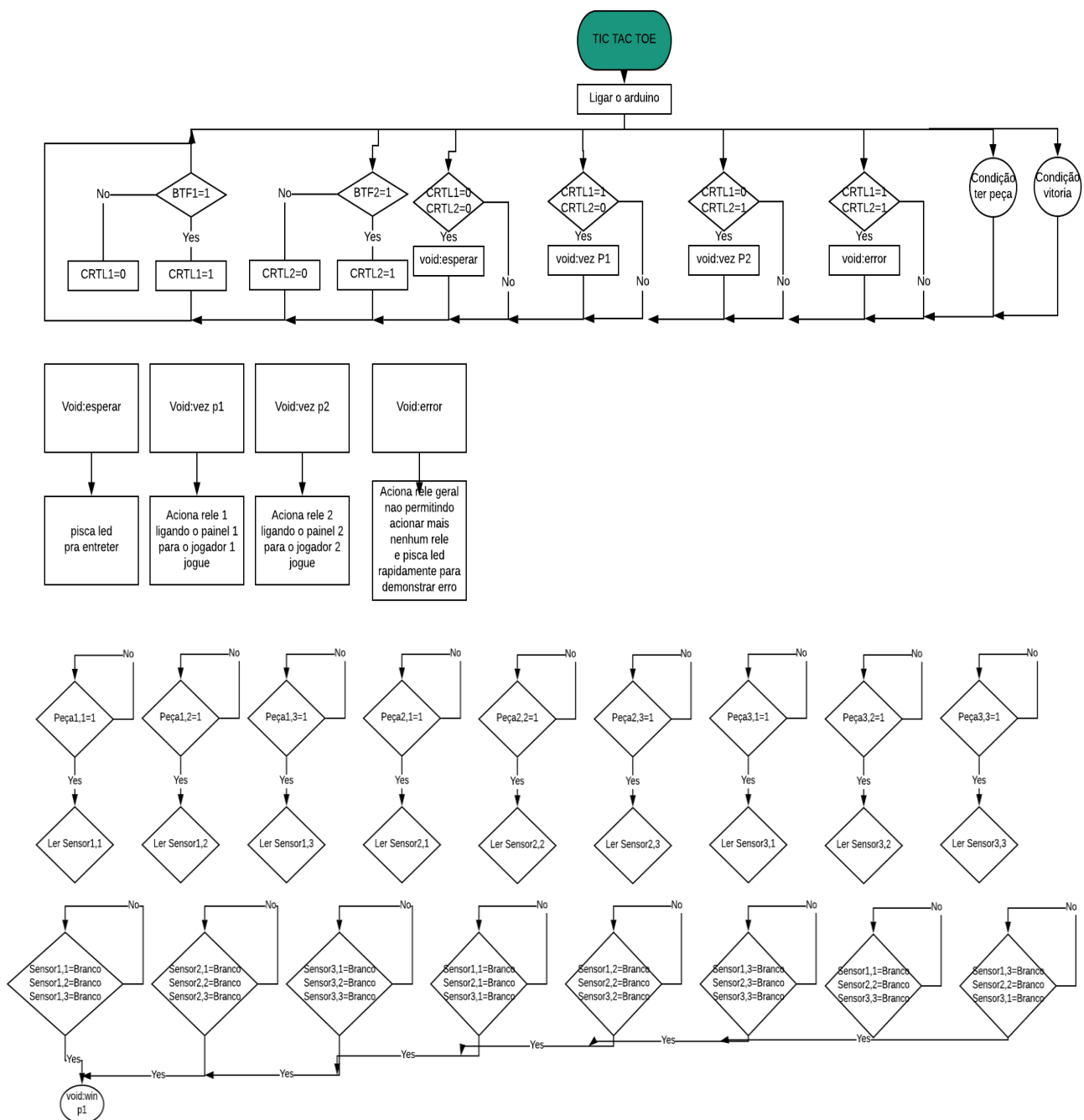


Figura 15 – Diagrama em blocos do processo

2.3 - Parte Lógica:

Esta parte é uma das mais importantes para o projeto, pois é responsável por conectar o funcionamento dos painéis, do intertravamento entre os turnos de cada jogador e o tabuleiro com seus sensores e LEDs com o objetivo de garantir uma jogabilidade fácil e automatizada.

- Fluxograma do Processo



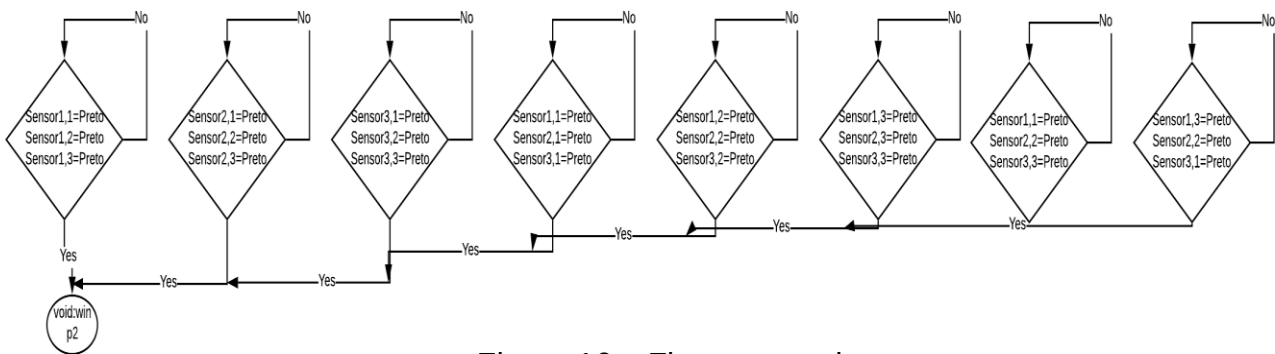


Figura 16 – Fluxograma do processo

Explicando os fluxogramas acima, basicamente o Arduino começará sua leitura com as chaves de turno de cada painel, com um loop de espera até que a chave do painel seja acionada. Uma vez ativada, o Arduino liberará o relé que alimenta o painel para que o jogador faça sua jogada. Ao término, o Arduino aguarda o jogador passar a vez para o outro com a chave. Caso os dois jogadores operem os painéis ao mesmo tempo, uma mensagem de erro será expressa por um LED piscante.

Já com a jogabilidade acontecendo, o Arduino ficará no aguardo de uma nova peça para se conectar ao sensor e identificar de que cor é aquela peça. Sem falar que existem todas as possibilidades de vitória tanto de combinações das peças pretas como das brancas, pois o Arduino armazena esses dados em sua programação junto das posições de matriz 3x3 que compõem cada casa do jogo da velha.

2.4 - Parte Mecânica:

- Croqui

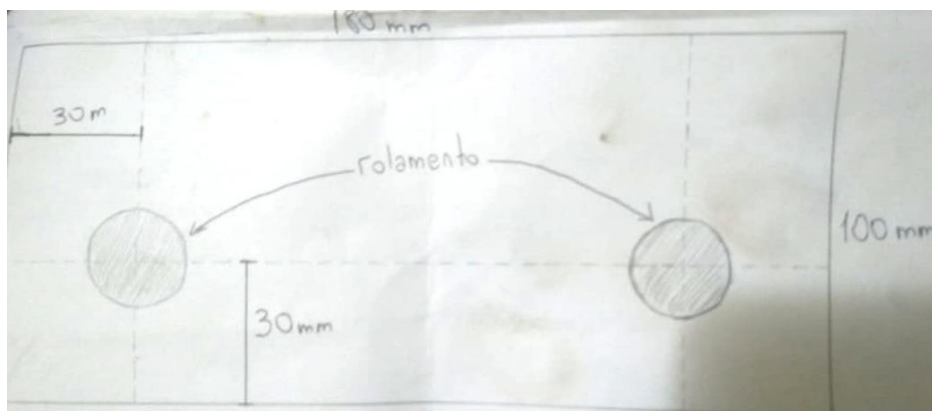


Figura 17 – Esboço do carro menor

Esta imagem retrata o esboço da estrutura do carro móvel menor, com seus respectivos rolamentos que mais tarde foram substituídos pela peça abaixo.

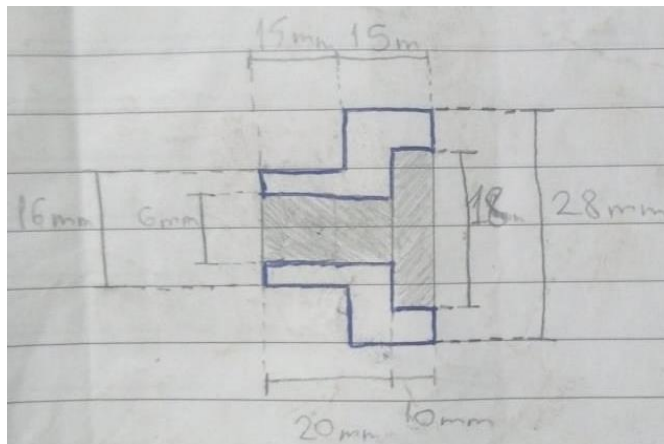


Figura 18 – Esboço da peça

- Desenho

Abaixo se segue o modelo da estrutura de aço do projeto, com o tabuleiro no centro.

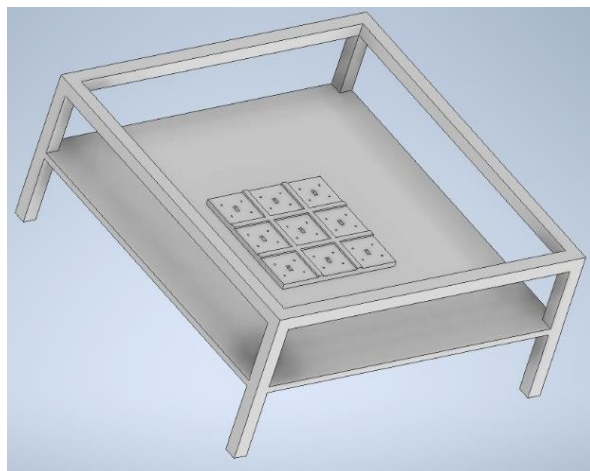


Figura 19 – Desenho da estrutura de aço

Logo em seguida, temos o desenho completo dos carros móveis menor e maior, com suas dimensões.

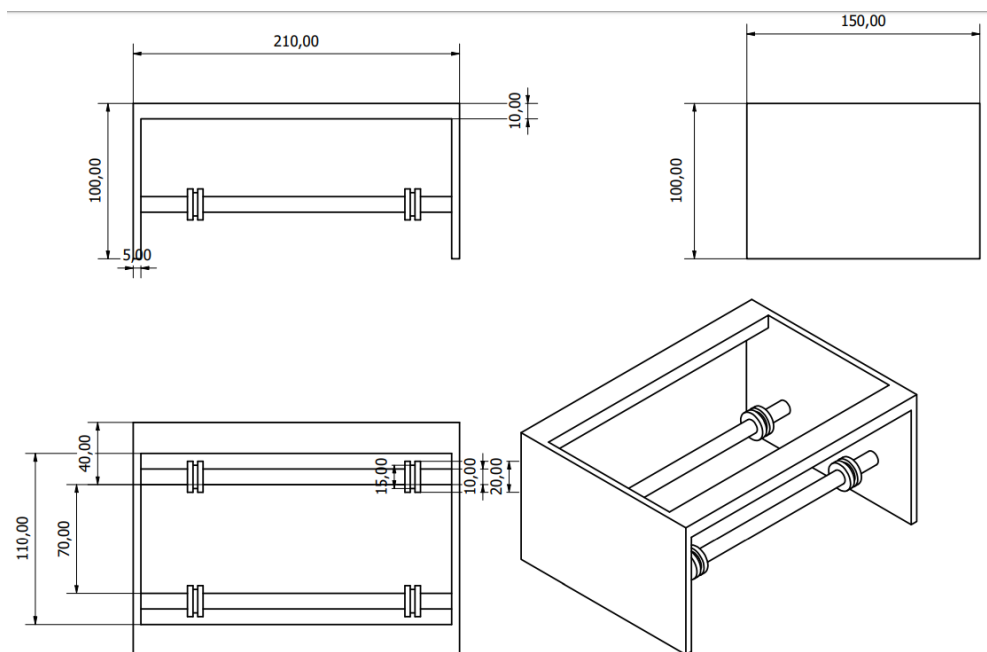


Figura 20 – Desenho do carro menor

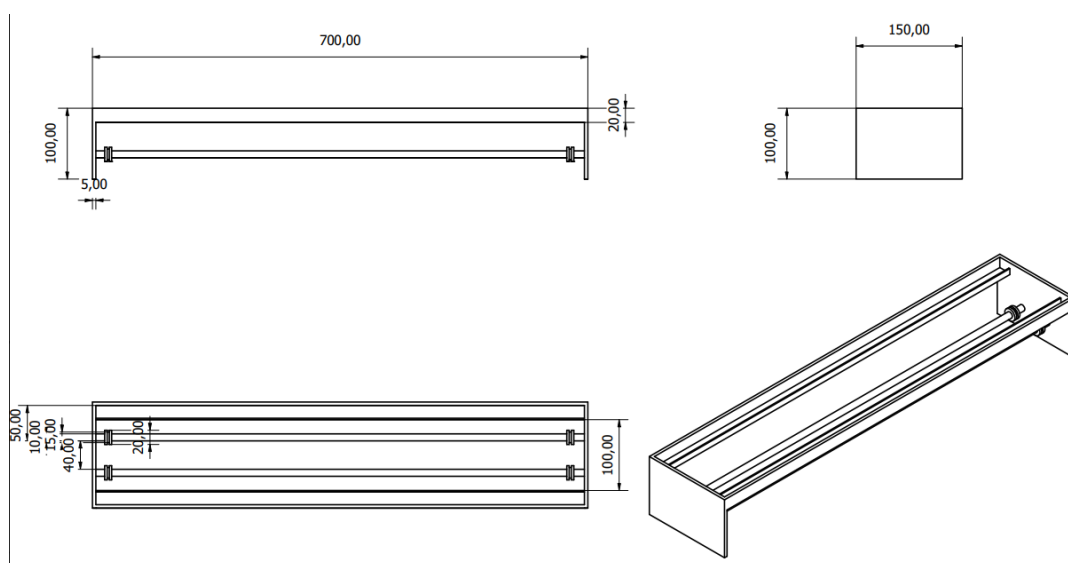


Figura 21 – Desenho do carro maior

- Pesquisa de Material

Percebemos que o uso do aço galvanizado seria fundamental para a proteção da estrutura, pois sua composição com o elemento zinco garante maior resistência ao material. Segundo o site Aramax, “O **aço galvanizado** é aquele que recebe um **revestimento de zinco**, que potencializa a proteção contra corrosão”

2.5 - Cronograma Geral e Cargos

CUSTO HORA-HOMEM										
EMPREGADO	SALÁRIO / HORA	HORAS / DIA	1º Turno			2º Turno			SALÁRIO/DIA	SALÁRIO FINAL MÊS
			ENTRADA	SAÍDA	TOTAL	ENTRADA	SAÍDA	TOTAL		
Eric	R\$ 12,50	08:00	08:00	12:30	04:30	13:30	17:00	03:30	R\$ 100,00	R\$ 2.000,00
Felipe	R\$ 12,50	08:00	08:00	12:30	04:30	13:30	17:00	03:30	R\$ 100,00	R\$ 2.000,00
Henry	R\$ 12,50	08:00	08:00	12:30	04:30	13:30	17:00	03:30	R\$ 100,00	R\$ 2.000,00
Luidgi	R\$ 13,09	08:00	08:00	12:30	04:30	13:30	17:00	03:30	R\$ 104,72	R\$ 2.094,40
Pedro	R\$ 12,50	08:00	08:00	12:30	04:30	13:30	17:00	03:30	R\$ 100,00	R\$ 2.000,00
Thiago Sales	R\$ 12,50	08:00	08:00	12:30	04:30	13:30	17:00	03:30	R\$ 100,00	R\$ 2.000,00
Thiago Yuiti	R\$ 13,10	08:00	08:00	12:30	04:30	13:30	17:00	03:30	R\$ 104,80	R\$ 2.096,00
Victor Vicente	R\$ 13,00	08:00	08:00	12:30	04:30	13:30	17:00	03:30	R\$ 104,00	R\$ 2.080,00
TOTAL FOLHA/DIA									R\$ 813,52	R\$ 16.270,40

Figura 22 – Custo Hora-Homem

Eric – Técnico de Programação – Agosto até início de Novembro

Felipe – Auxiliar Mecânico – Agosto até final de Outubro

Henry – Técnico Mecânico - Agosto até final de Outubro

Luidgi – Responsável Mecânico - Agosto até final de Outubro

Pedro – Auxiliar de Almoxarifado - Agosto até final de Outubro

Thiago S. – Técnico Elétrico – Início de Setembro até Novembro

Thiago Y. – Responsável de Programação – Agosto até início de Novembro

V. Vicente – Responsável Elétrico - Início de Setembro até Novembro

3 – Desenvolvimento do Projeto

Esta etapa retrata todo o desenvolvimento do Tic Tac Toe, a estrutura mecânica e como fizemos suas integrações software-hardware.

3.1 – Desenvolvimento da estrutura

Inicialmente, o projeto Tic Tac Toe precisava de uma estrutura maior não apenas para garantir melhor dinâmica de movimento da ponte rolante, mas também para melhorar a experiência de jogo em relação ao protótipo que fora realizado por nós para a 48ª EXCUTE. Para tanto, fizemos uma estrutura em forma quadrada para dispor o tabuleiro no centro. Utilizamos algumas barras de aço galvanizado que posteriormente foram medidas e cortadas para formarem o “esqueleto” deste jogo da velha tecnológico. Seguindo este objetivo, a junção das barras de aço foi realizada com a máquina de solda que se encontra disponível na oficina escolar.



Figura 23 – Estrutura Tic Tac Toe recém finalizada.



Figura 24 – instrumento de solda utilizado para confeccionar toda a parte mecânica.



Figura 25 – acabamento das barras da estrutura com utilização de esmeril.

3.2 – Desenvolvimento dos carros moveis

Logo após o desenvolvimento da estrutura, ficamos de solucionar o problema da baixa velocidade da ponte rolante do protótipo, o que poderia causar uma jogabilidade maçante. Depois de alguns testes, percebemos que o passo da rosca que girava em torno do eixo do motor era muito desproporcional ao mesmo, o que acabava retardando o movimento do carro móvel. No teste passado, a rosca – presa por uma solda ao carro movia-o através de sua rotação por um eixo acoplado ao motor. Para solucionar o problema, decidimos realizar o movimento confeccionando um carro com rodas, que uma vez presas ao eixo do motor girarão sobre um trilho de aço soldado na estrutura, proporcionando o movimento dos carros para a correta condução das peças. A confecção dos carros também foi realizada com solda.

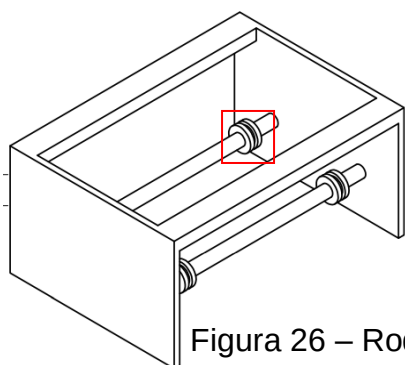


Figura 26 – Roda do carro menor

Note na figura ao lado as rodas em destaque que serão encaixadas no trilho disposto no carro maior.

Desta forma, o movimento dos dois eixos X e Y será facilmente conectado ao jogo.

Em seguida (figura 27), vemos os carros recém confeccionados e prontos para serem acoplados a estrutura. Perceba que o carro menor já está com suas

rodas conectadas ao trilho do carro maior, enquanto este último terá suas rodas (em destaque na figura 29) conectadas ao trilho da estrutura.



Figura 27 – Carros Prontos



Figura 28 – Vista dos carros
na estrutura

A figura acima retrata nosso primeiro teste para verificar a posição dos carros na estrutura. Nele, conseguimos estabelecer que o carro menor iria se movimentar em X e o maior em Y.

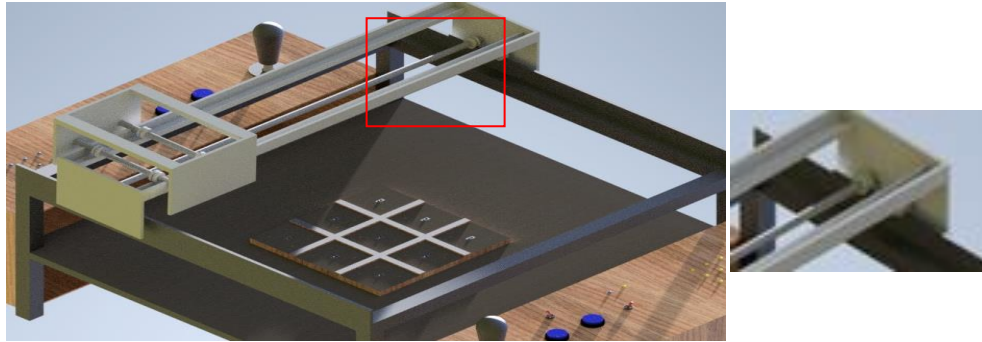


Figura 29 – estado do carro maior na estrutura final.

Nos testes finais dos carros, decidimos trocar os rolamentos que unem o motor com o eixo por uma peça torneada, feita sob medida para o encaixe no motor e no eixo. O objetivo é garantir uma maior estabilidade.



Figura 30 – Peça torneada

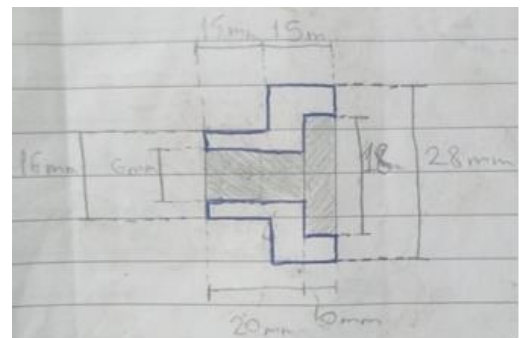


Figura 31 – Esboço da peça

Para produzir esta peça, utilizamos um torno industrial de cunho profissional para realizar os furos e os devidos cortes. Foi necessário a utilização do mesmo para a fabricação das rodas dos dois carros.

Ao final, realizamos o teste definitivo de velocidade.



Figura 32 – Teste do carro em movimento. Observa-se o motor montado com o eixo onde se encontram as rodas.

3.3 – Desenvolvimento dos painéis

Para ser possível de se jogar, é vero que o jogo da velha automatizado necessita de controladores e botões para movimentar as peças, além de uma plataforma acessível integrada a estrutura onde estes últimos serão localizados. Para tanto, a produção de painéis foi necessária. Com objetivo de proporcionar um bom acabamento e melhor custo-benefício, os painéis de controle foram confeccionados com madeira em MDF sendo cada parte devidamente medida para posteriormente ser submetida ao corte a laser.

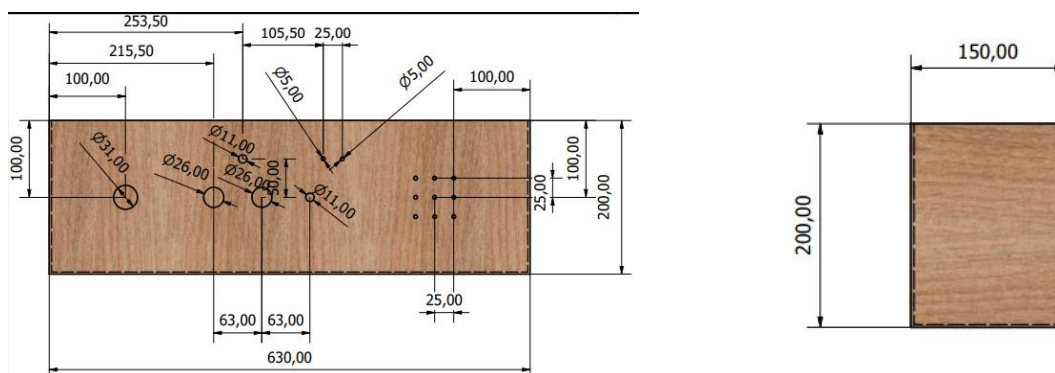


Figura 33 – Cortes do painel

Cada componente foi medido utilizando um paquímetro, e os cortes foram dispostos sobre o painel de acordo com seu diâmetro para fácil introdução dos controladores e botões (figura 34).

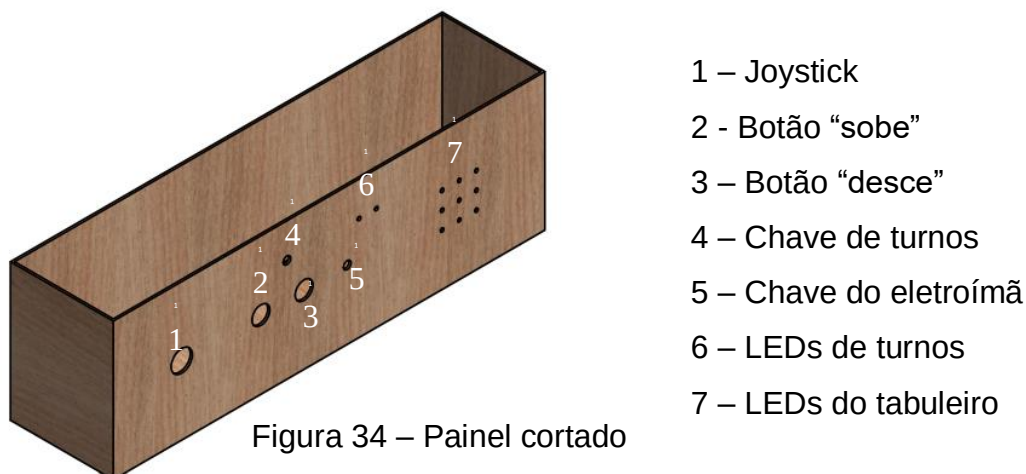


Figura 34 – Painel cortado



Figura 35 – Painel pronto

A figura acima mostra um dos painéis prontos com todos os componentes introduzidos na estrutura de MDF.

Os painéis foram montados utilizando cantoneiras e parafusando-as em cada peça de MDF.



Figura 36 – Montagem do painel com parafusos

Realizamos os primeiros testes e conseguimos estabelecer uma boa posição dos painéis em relação a estrutura.

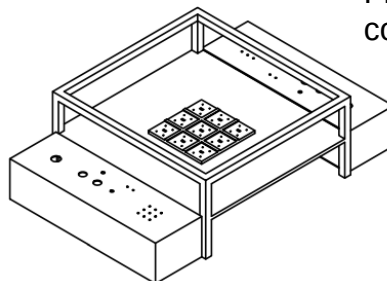


Figura 37 – Painéis conectados à estrutura.

3.4 – Desenvolvimento do tabuleiro

Como nossa estrutura possui o interior quadrado, medimos o tabuleiro a ser feito com as dimensões da estrutura. Porém para garantir que o tabuleiro fosse introduzido facilmente nos suportes, foi necessário subtrair a área dos 4 cantos, que seriam as “pernas” da estrutura. Assim, com as medidas corretas o tabuleiro pode ser introduzido e preso com segurança na estrutura.

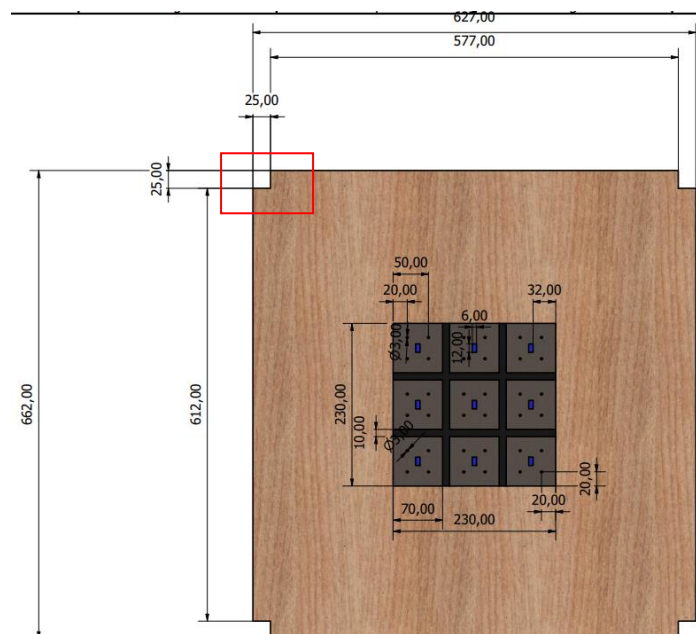


Figura 38 – Tabuleiro com os cantos de encaixe em destaque

Note também que o tabuleiro tem dimensões especiais para a área de cada casa que compõe o jogo da velha e também possui espaços para a introdução dos sensores ópticos (na figura 38, são os pontos azuis) e para a passagem dos cabos de presença.

A confecção do tabuleiro também foi integralmente produzida através de corte a laser.

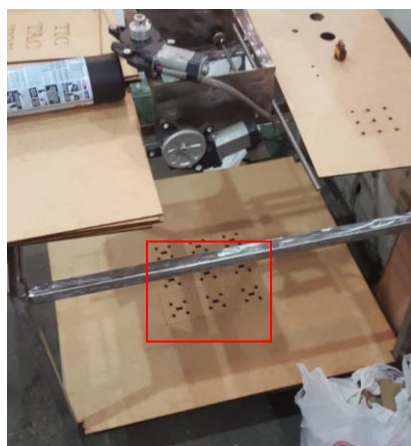


Figura 39 – Estrutura com o tabuleiro destacado

3.5 – Desenvolvimento da parte elétrica/eletrônica

Primeiramente, tivemos que produzir uma placa de circuito impresso. Nela fica o circuito do tabuleiro, onde serão conectados à entrada do Arduino os sensores ópticos e os cabos de presença (veja circuito na página 15).

Baseado no desenho técnico, transportamos com auxílio de uma caneta “marca CD”, o circuito para a placa de fenolite adquirida.

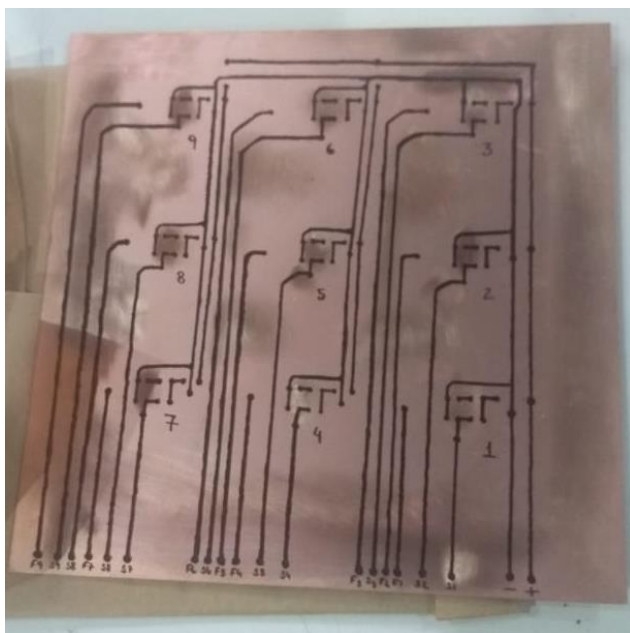


Figura 40 – Placa com circuito em caneta

Percebe-se que os números de 1 a 9 representam as casas do jogo da velha e em cada uma delas se encontram os furos onde serão soldados os sensores e os cabos de presença.

Após isso, com a utilização segura de percloroeto de ferro misturado com água, submergimos a placa e esperamos a corrosão da parte de cobre inutilizável.

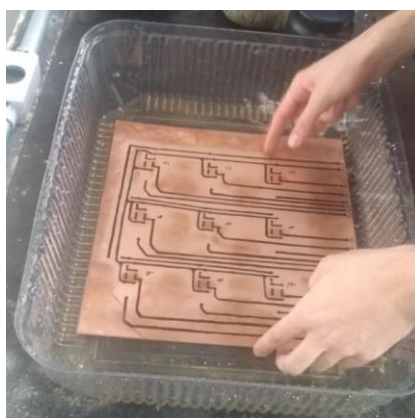


Figura 41 – Placa submersa em água

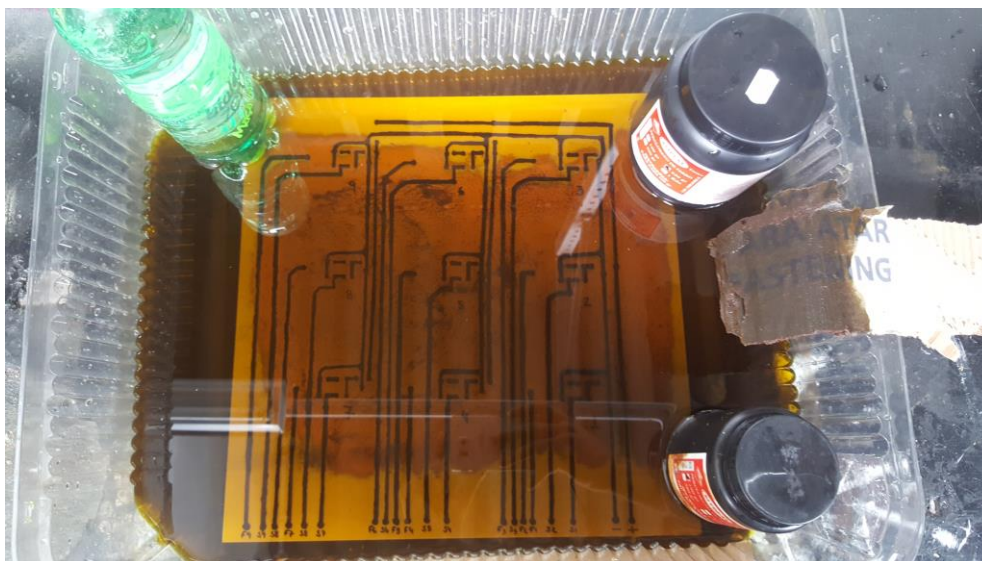


Figura 42 – Placa submersa em percloroeto de ferro e água

Após a corrosão a trilha de tinta se transformou em trilha de cobre, pronta para conduzir eletricidade. Furamos cada marcação com uma broca de 1mm e em seguida, soldamos os componentes utilizando ferro de solda junto de uma reserva de estanho. A figura abaixo mostra onde soldamos o sensor óptico – note que cada um desfruta de alimentação negativa e uma trilha separada que será conectada na entrada do Arduino.

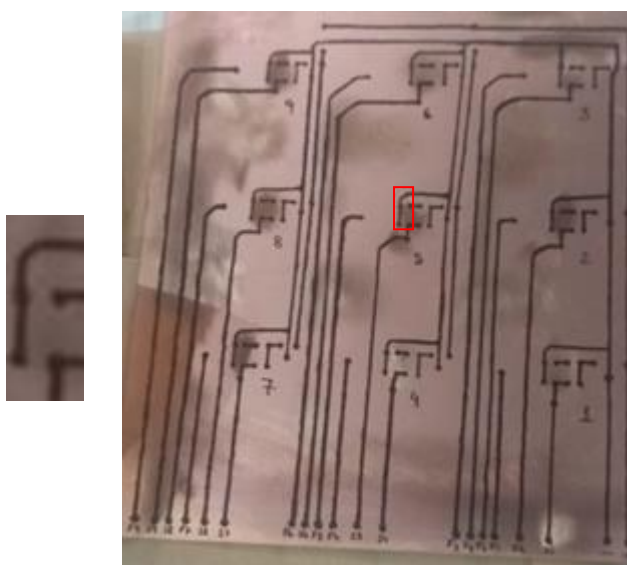


Figura 43 – Localização da solda do sensor na placa

Cada sensor possui a necessidade de ser alimentado com uma tensão positiva de 5V oriunda do Arduino. Para isso, fizemos no circuito da placa conexões com jumpers para puxar a eletricidade da trilha positiva para a trilha dos sensores.

Para a proteção dos LEDs infravermelhos do sensor, não esquecemos de soldar os dois resistores em seu circuito.

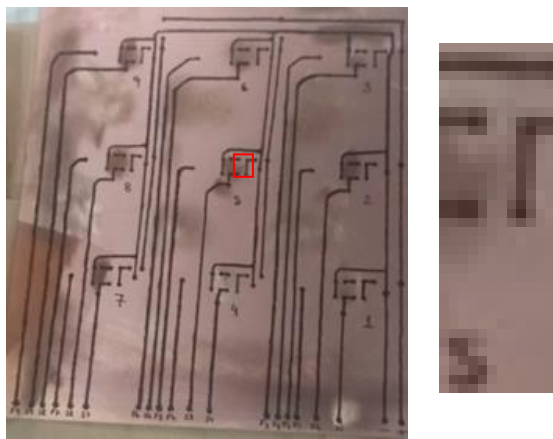


Figura 44 – Localização de solda dos dois resistores

Por último, soldamos um cabo em cada trilha de entrada para levar o sinal até o Arduino. Isto inclui o negativo que vem do Arduino (pino GROUND) e o positivo de 5V do mesmo. A placa ficara abaixo do tabuleiro, deixando o sensor e o cabo de presença em contato com as peças de cima.

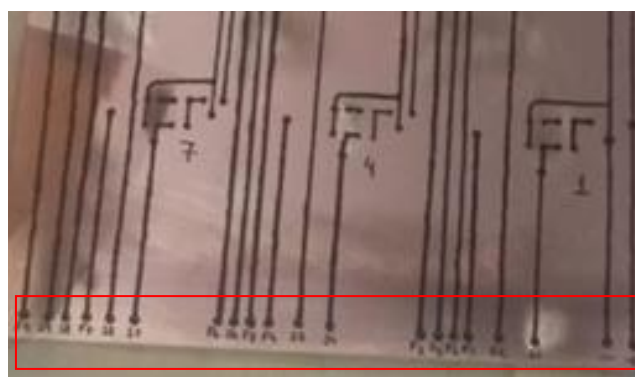
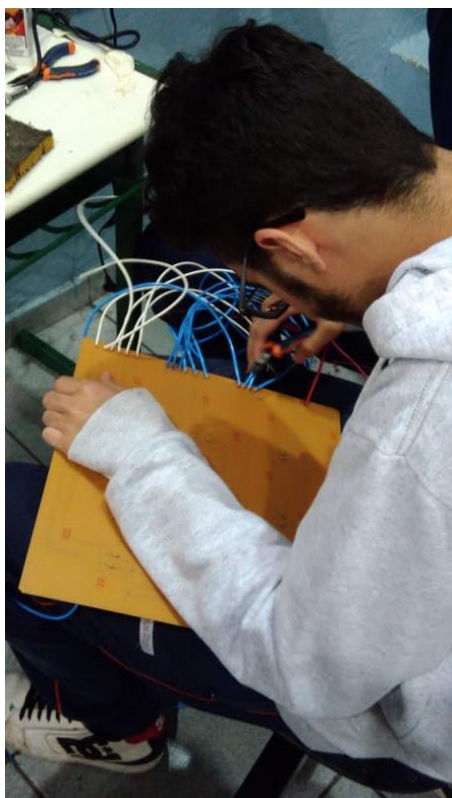


Figura 45 – Trilhas dos sinais de entrada entre placa e Arduino

Figura 46 – Cabos de entrada soldados na placa

Utilizando parafusos, prendemos no painel, os três relés que são responsáveis por liberar o circuito de movimento (joystick, botões, etc.). Logo após, soldamos cabos para conecta-los na saída do Arduino (veja circuitos das págs. 18 e 19).

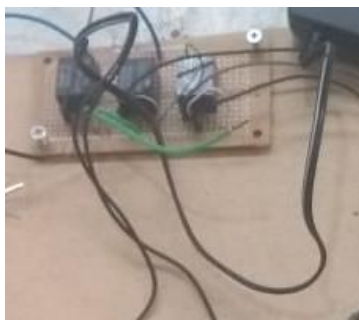


Figura 47 – Relés geral e de painel

Os LEDs de posição das peças no painel foram soldados com cabos em paralelo para compartilhar a alimentação negativa do Arduino, além de deixarmos separados os cabos para alimentação positiva.



Figura 48 – Solda dos cabos nos LEDs

Em seguida, soldamos mais cabos nas extremidades dos joysticks, botões e eletroímã, que constituem o painel de controle de cada jogador (figuras 49 e 50). Soldamos fios com alimentação positiva de 12V - vindas dos relés de saída do Arduino - em paralelo com essas entradas para serem conectadas à interface de potência dos 6 relés que controlam a inversão de rotação dos motores (figura 51).



Figura 49 – Solda dos cabos no joystick



Figura 50 – solda dos cabos nos botões e chaves



Figura 51 – interface de potência

A interface de potência é alimentada com uma fonte de alimentação 12V que depois interligamos com cabos aos motores da ponte.

Para o transporte das peças, usamos um eletroímã conectado a uma chave presente em cada painel. Nesta parte soldamos diretamente entrada e saída: chave e eletroímã.

3.6 – Desenvolvimento da Programação

Para garantir que não haveria erros na programação, os primeiros testes foram feitos em um protoboard. Os sensores ópticos eram conectados ao Arduino com a ajuda desta placa e os programadores obtiveram sucesso nos testes sem precisar do tabuleiro de antemão.

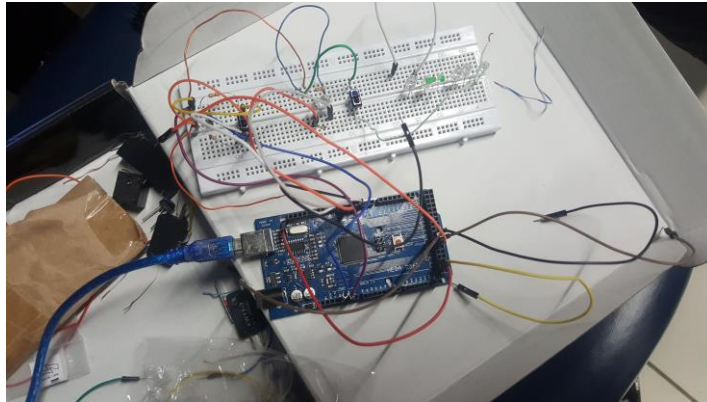


Figura 52 – Teste com protoboard



Figura 53 – Desenvolvimento da programação

Na parte logica, precisávamos encontrar uma forma do programa entender que o jogo da velha possui 9 posições, para facilitar a identificação de uma rodada vencida. Para isso, adotamos o sistema de uma matriz 3x3(figura 54).

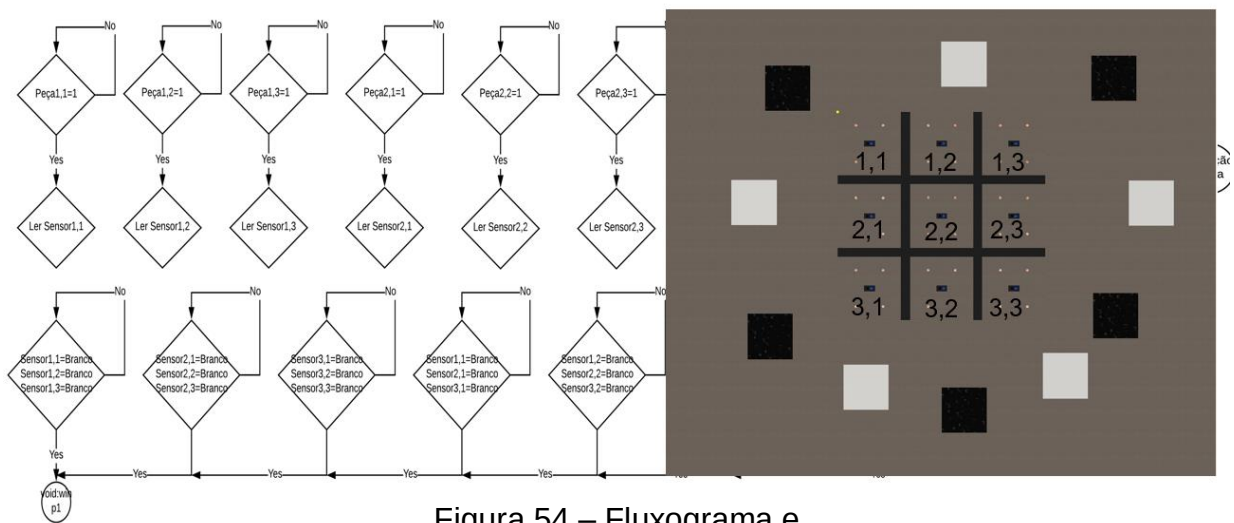


Figura 54 – Fluxograma e matriz 3x3

De acordo com o fluxograma planejado (veja pag. 22), programamos o Arduino para identificar as peças nas posições (1,1; 1,2; etc.), através dos cabos de cobre que conduzirão eletricidade com o material condutor das peças. Em seguida, o sensor óptico irá identificar a cor da peça e então mandar o sinal para o Arduino. Se o Arduino identificar com ajuda do sensor que existem, por exemplo, 3 peças brancas seguidas (figura 55), a rodada é ganha pelo jogador dono das peças brancas. Esta é apenas uma das possibilidades de vitória pré-estabelecidas, pois nossa lógica possui todos os possíveis casos, como mostrado no fluxograma.

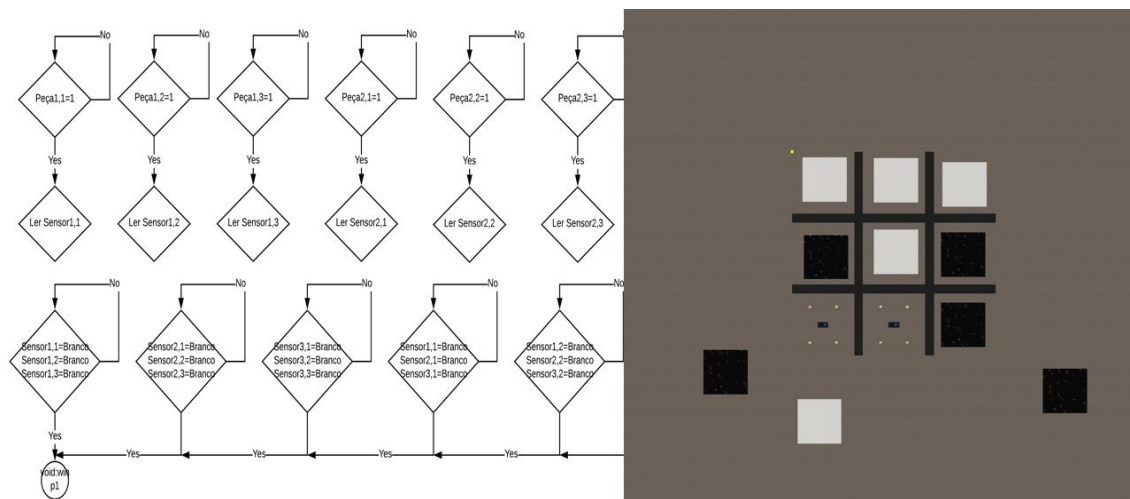


Figura 55 – Fluxograma e exemplo de vitória

Além de organizar o microcontrolador Arduino Mega com suas respectivas entradas e saídas, é preciso realizar o desenvolvimento das linhas de programação.

Para tanto, usamos o aplicativo de programação Arduino CC, para armazenar os dados. Abaixo, seguem as linhas de programação, com alguns comentários para melhor compreensão:

```

/////Programação TIC TAC TOE
#define BP1_f 45// botão passa vez fisico do jogador 1
int BP1_l = 0;//Botão jogador 1 lógico
#define BP2_f 46// botão passavez fisico do jogador 2
int BP2_l = 0;//botão jogador 2 lógico

int CTRL1 = 0;//COMEÇAR A VARIÁVEL EM 0 DE CONTROLE DO
JOGADOR 1
int CTRL2 = 0;//COMEÇAR A VARIÁVEL EM 0 DE CONTROLE DO
JOGADOR 2

int mP11 = 0;//MEMORIA SE TEM PEÇA EM 1,1
int P11 = 0; //variavel de controle se tem peça no 1,1

int mP12 = 0;//MEMORIA SE TEM PEÇA EM 1,2
int P12 = 0; // variavel de controle se tem peça no 1,2

int mP13 = 0;//MEMORIA SE TEM PEÇA EM 1,3
int P13 = 0; //variavel de controle se tem peça no 1,3

int mP21 = 0;//MEMORIA SE TEM PEÇA EM 2,1
int P21 = 0; // variavel de controle se tem peça no 2,1

int mP22 = 0;
int P22 = 0; //variavel de controle se tem peça no 2,2

int mP23 = 0;
int P23 = 0; // variavel de controle se tem peça no 2,3

int mP31 = 0;

```

```
int P31 = 0; //variavel de controle se tem peça no 3,1
```

```
int mP32 = 0;
```

```
int P32 = 0; // variavel de controle se tem peça no 3,2
```

```
int mP33 = 0;
```

```
int P33 = 0; //variavel de controle se tem peça no 3,3
```

```
#define LP1 1
```

```
#define LP2 2
```

```
#define STRCT11 3 //Sensor TRCT p/ 1,1
```

```
#define SPECA11 4 //Sensor Posicao 1,1
```

```
#define LB11 5 //Led peça branca 1,1
```

```
#define LP11 6 //led peça preta 1,1
```

```
#define STRCT12 7 //Sensor TRCT p/ 1,2
```

```
#define SPECA12 8 //Sensor Posicao 1,2
```

```
#define LB12 9 //Led peça branca 1,2
```

```
#define LP12 10 //led peça preta 1,2
```

```
#define STRCT13 11 //Sensor TRCT p/ 1,3
```

```
#define SPECA13 12 //Sensor Posicao 1,3
```

```
#define LB13 13 //Led peça branca 1,3
```

```
#define LP13 14 //led peça preta 1,3
```

```
#define STRCT21 15 //Sensor TRCT p/ 2,1
```

```
#define SPECA21 16 //Sensor Posicao 2,1
```

```
#define LB21 17 //Led peça branca 2,1
```

```
#define LP21 18 //led peça preta 2,1
```

```
#define STRCT22 19 //Sensor TRCT p/ 2,2
```

```
#define SPECA22 20 //Sensor Posicao 2,2
```

```
#define LB22 21 //Led peça branca 2,2
```

```
#define LP22 22 //led peça preta 2,2
```

```
#define STRCT23 23 //Sensor TRCT p/ 2,3
#define SPECA23 24 //Sensor Posicao 2,3
#define LB23 25 //Led peca branca 2,3
#define LP23 26 //led peca preta 2,3
```

```
#define STRCT31 27 //Sensor TRCT p/ 3,1
#define SPECA31 28 //Sensor Posicao 3,1
#define LB31 29 //Led peca branca 3,1
#define LP31 30 //led peca preta 3,1
```

```
#define STRCT32 31 //Sensor TRCT p/ 3,2
#define SPECA32 32 //Sensor Posicao 3,2
#define LB32 33 //Led peca branca 3,2
#define LP32 34 //led peca preta 3,2
```

```
#define STRCT33 35 //Sensor TRCT p/ 3,3
#define SPECA33 36 //Sensor Posicao 3,3
#define LB33 37 //Led peca branca 3,3
#define LP33 38 //led peca preta 3,3
```

```
const int rele1 = 39;
const int rele2 = 41;
const int releG = 50;
void setup(){
  Serial.begin(9600);
  CTRL1 = 0;
  CTRL2 = 0;
  pinMode(SPECA11, INPUT_PULLUP);

  pinMode(SPECA12, INPUT_PULLUP);

  pinMode(SPECA13, INPUT_PULLUP);

  pinMode(SPECA21, INPUT_PULLUP);
```

```
pinMode(SPECA22, INPUT_PULLUP);
```

```
pinMode(SPECA23, INPUT_PULLUP);
```

```
pinMode(SPECA31, INPUT_PULLUP);
```

```
pinMode(SPECA32, INPUT_PULLUP);
```

```
pinMode(SPECA33, INPUT_PULLUP);
```

```
pinMode(BP1_f, INPUT_PULLUP);
```

```
pinMode(BP2_f, INPUT_PULLUP);
```

```
pinMode(releG, OUTPUT);
```

```
pinMode(rele1, OUTPUT);
```

```
pinMode(rele2, OUTPUT);
```

```
pinMode(LP1, OUTPUT);
```

```
pinMode(LP2, OUTPUT);
```

```
pinMode(LP11, OUTPUT);
```

```
pinMode(LB11, OUTPUT);
```

```
pinMode(LP12, OUTPUT);
```

```
pinMode(LB12, OUTPUT);
```

```
pinMode(LP13, OUTPUT);
```

```
pinMode(LB13, OUTPUT);
```

```
pinMode(LP21, OUTPUT);
```

```
pinMode(LB21, OUTPUT);
```

```
pinMode(LP22, OUTPUT);
```

```
pinMode(LB22, OUTPUT);
```

```
pinMode(LP23, OUTPUT);
```

```
pinMode(LB23, OUTPUT);
```

```
pinMode(LP31, OUTPUT);
```

```
pinMode(LB31, OUTPUT);
```

```
pinMode(LP32, OUTPUT);
```

```

pinMode(LB32, OUTPUT);
pinMode(LP33, OUTPUT);
pinMode(LB33, OUTPUT);

}

void loop(){
digitalWrite(releG,LOW);
////////Controlador de vez////////
if(CTRL1 == 1 && CTRL2 == 0){
    vezP1();
}
else if(CTRL1 == 0 && CTRL2 == 1){
    vezP2();
}
else if(CTRL1 == 0 && CTRL2 == 0){
    piscapiscanatalino();
}
else {
    erro();
}

BP1_I = digitalRead(BP1_f);
if(BP1_I == 0){
    CTRL1 = 1;
}
else{
    CTRL1 = 0;
}

BP2_I = digitalRead(BP2_f);
if(BP2_I == 0){
    CTRL2 = 1;
}
else{
    CTRL2 = 0;
}

```



```

if(P12 == LOW){
    if( (digitalRead(STRCT12)) == HIGH){
        Serial.println("PEÇA PRETA 1,2");  mP12 = 1;
    }
    else{
        Serial.println("PEÇA BRANCA 1,2"); mP12 = 2;
    }
}
else if(P12 == HIGH){
    Serial.println("P12 SEM PECA"); mP12 = 0;
}

if(mP12 == 1){
    digitalWrite(LP12,HIGH);
    digitalWrite(LB12,LOW);
}
else if (mP12 == 2){
    digitalWrite(LB12,HIGH);
    digitalWrite(LP12,LOW);
}
else if(mP12 == 0){
    digitalWrite(LB12,LOW);
    digitalWrite(LP12,LOW);
}

/////////////////////////////////////////////////////////////////

//le Posicao 1,3/////////////////////////////////////////////////////////////////
P13 = digitalRead(SPECA13);
if(P13 == LOW){
    if( (digitalRead(STRCT13)) == HIGH){
        Serial.println("PEÇA PRETA 1,3");  mP13 = 1;
    }
    else{
        Serial.println("PEÇA BRANCA 1,3"); mP13 = 2;
    }
}

```



```

    }
}
else if(P13 == HIGH){
    Serial.println("P13 SEM PECA"); mP13 = 0;
}

if(mP13 == 1){
    digitalWrite(LP13,HIGH);
    digitalWrite(LB13,LOW);
}
else if (mP13 == 2){
    digitalWrite(LB13,HIGH);
    digitalWrite(LP13,LOW);
}
else if(mP13 == 0){
    digitalWrite(LB13,LOW);
    digitalWrite(LP13,LOW);
}

/////////////////////////////////////////////////////////////////

//le Posicao 2,1/////////////////////////////////////////////////////////////////
P21 = digitalRead(SPECA21);
if(P21 == LOW){
    if( (digitalRead(STRCT21)) == HIGH){
        Serial.println("PEÇA PRETA 2,1");  mP21 = 1;
    }
    else{
        Serial.println("PEÇA BRANCA 2,1"); mP21 = 2;
    }
}
else if(P21 == HIGH){
    Serial.println("P21 SEM PECA"); mP21 = 0;
}

```

```

if(mP21 == 1){
    digitalWrite(LP21,HIGH);
    digitalWrite(LB21,LOW);
}
else if (mP21 == 2){
    digitalWrite(LB21,HIGH);
    digitalWrite(LP21,LOW);
}
else if(mP21 == 0){
    digitalWrite(LB21,LOW);
    digitalWrite(LP21,LOW);
}

/////////////////////////////////////////////////////////////////

//le Posicao 2,2/////////////////////////////////////////////////////////////////
P22 = digitalRead(SPECA22);
if(P22 == LOW){
    if( (digitalRead(STRCT22)) == HIGH){
        Serial.println("PEÇA PRETA 2,2"); mP22 = 1;
    }
    else{
        Serial.println("PEÇA BRANCA 2,2"); mP22 = 2;
    }
}
else if(P22 == HIGH){
    Serial.println("P22 SEM PECA"); mP22 = 0;
}

if(mP22 == 1){
    digitalWrite(LP22,HIGH);
    digitalWrite(LB22,LOW);
}
else if (mP22 == 2){
    digitalWrite(LB22,HIGH);

```

```

    digitalWrite(LP22,LOW);
}
else if(mP22 == 0){
    digitalWrite(LB22,LOW);
    digitalWrite(LP22,LOW);
}
/////////////////////////////////////////////////////////////////

//le Posicao 2,3/////////////////////////////////////////////////////////////////
P23 = digitalRead(SPECA23);
if(P23 == LOW){
    if( (digitalRead(STRCT23)) == HIGH){
        Serial.println("PEÇA PRETA 2,3");  mP23 = 1;
    }
    else{
        Serial.println("PEÇA BRANCA 2,3");  mP23 = 2;
    }
}
else if(P23 == HIGH){
    Serial.println("P23 SEM PECA"); mP23 = 0;
}

if(mP23 == 1){
    digitalWrite(LP23,HIGH);
    digitalWrite(LB23,LOW);
}
else if (mP23 == 2){
    digitalWrite(LB23,HIGH);
    digitalWrite(LP23,LOW);
}
else if(mP23 == 0){
    digitalWrite(LB23,LOW);
    digitalWrite(LP23,LOW);
}

```

```
////////////////////////////////////////////////////////////////
```

```
//le Posicao 3,1////////////////////////////////////////////////////////////////
```

```
P31 = digitalRead(SPECA31);
if(P31 == LOW){
    if( (digitalRead(STRCT31)) == HIGH){
        Serial.println("PEÇA PRETA 3,1");  mP31 = 1;
    }
    else{
        Serial.println("PEÇA BRANCA 3,1"); mP31 = 2;
    }
}
else if(P31 == HIGH){
    Serial.println("P31 SEM PECA"); mP31 = 0;
}
```

```
if(mP31 == 1){
    digitalWrite(LP31,HIGH);
    digitalWrite(LB31,LOW);
}
else if (mP31 == 2){
    digitalWrite(LB31,HIGH);
    digitalWrite(LP31,LOW);
}
else if(mP31 == 0){
    digitalWrite(LB31,LOW);
    digitalWrite(LP31,LOW);
}
```

```
////////////////////////////////////////////////////////////////
```

```
//le Posicao 3,2////////////////////////////////////////////////////////////////
```

```
P32 = digitalRead(SPECA32);
if(P32 == LOW){
    if( (digitalRead(STRCT32)) == HIGH){
        Serial.println("PEÇA PRETA 3,2");  mP32 = 1;
```

```

    }
    else{
        Serial.println("PEÇA BRANCA 3,2"); mP32 = 2;
    }
}
else if(P32 == HIGH){
    Serial.println("P32 SEM PECA"); mP32 = 0;
}

if(mP32 == 1){
    digitalWrite(LP32,HIGH);
    digitalWrite(LB32,LOW);
}
else if (mP32 == 2){
    digitalWrite(LB32,HIGH);
    digitalWrite(LP32,LOW);
}
else if(mP32 == 0){
    digitalWrite(LB32,LOW);
    digitalWrite(LP32,LOW);
}

/////////////////////////////////////////////////////////////////

//le Posicao 3,3/////////////////////////////////////////////////////////////////
P33 = digitalRead(SPECA33);
if(P33 == LOW){
    if( (digitalRead(STRCT33)) == HIGH){
        Serial.println("PEÇA PRETA 3,3"); mP33 = 1;
    }
    else{
        Serial.println("PEÇA BRANCA 3,3"); mP33 = 2;
    }
}
else if(P33 == HIGH){

```

```

        Serial.println("P33 SEM PECA"); mP33 = 0;
    }

    if(mP33 == 1){
        digitalWrite(LP33,HIGH);
        digitalWrite(LB33,LOW);
    }
    else if(mP33==2){
        digitalWrite(LB33,HIGH);
        digitalWrite(LP33,LOW);
    }
    else if(mP33 == 0){
        digitalWrite(LB33,LOW);
        digitalWrite(LP33,LOW);
    }

    //Condição de vitória 1 do jogador 1
    if ((mP11==1)&&(mP12==1)&&(mP13==1)){
        winP1();
        digitalWrite(LB11,HIGH);
        digitalWrite(LB12,HIGH);
        digitalWrite(LB13,HIGH);
        delay(200);
        digitalWrite(LB11,LOW);
        digitalWrite(LB12,LOW);
        digitalWrite(LB13,LOW);
        delay(200);
    }

    //Condição de vitória 2 do jogador 1
    if ((mP21==1)&&(mP22==1)&&(mP23==1)){
        winP1();
        digitalWrite(LB21,HIGH);
        digitalWrite(LB22,HIGH);
        digitalWrite(LB23,HIGH);
        delay(200);
    }

```

```

digitalWrite(LB21,LOW);
digitalWrite(LB22,LOW);
digitalWrite(LB23,LOW);
delay(200);
}
//Condição de vitória 3 do jogador 1
if ((mP31==1)&&(mP32==1)&&(mP33==1)){
winP1();
digitalWrite(LB31,HIGH);
digitalWrite(LB32,HIGH);
digitalWrite(LB33,HIGH);
delay(200);
digitalWrite(LB31,LOW);
digitalWrite(LB32,LOW);
digitalWrite(LB33,LOW);
delay(200);
}
//Condição de vitória 4 do jogador 1
if ((mP11==1)&&(mP21==1)&&(mP31==1)){
winP1();
digitalWrite(LB11,HIGH);
digitalWrite(LB21,HIGH);
digitalWrite(LB31,HIGH);
delay(200);
digitalWrite(LB11,LOW);
digitalWrite(LB21,LOW);
digitalWrite(LB31,LOW);
delay(200);
}
//Condição de vitória 5 do jogador 1
if ((mP12==1)&&(mP22==1)&&(mP32==1)){
winP1();
digitalWrite(LB12,HIGH);
digitalWrite(LB22,HIGH);

```

```

digitalWrite(LB32,HIGH);
delay(200);
digitalWrite(LB12,LOW);
digitalWrite(LB22,LOW);
digitalWrite(LB32,LOW);
delay(200);
}
//Condição de vitória 6 do jogador 1
if ((mP13==1)&&(mP23==1)&&(mP33==1)){
winP1();
digitalWrite(LB13,HIGH);
digitalWrite(LB23,HIGH);
digitalWrite(LB33,HIGH);
delay(200);
digitalWrite(LB13,LOW);
digitalWrite(LB23,LOW);
digitalWrite(LB33,LOW);
delay(200);
}
//Condição de vitória 7 do jogador 1
if ((mP11==1)&&(mP22==1)&&(mP33==1)){
winP1();
digitalWrite(LB11,HIGH);
digitalWrite(LB22,HIGH);
digitalWrite(LB33,HIGH);
delay(200);
digitalWrite(LB11,LOW);
digitalWrite(LB22,LOW);
digitalWrite(LB33,LOW);
delay(200);
}
//Condição de vitória 8 do jogador 1
if ((mP13==1)&&(mP22==1)&&(mP31==1)){
winP1();

```



```

digitalWrite(LB13,HIGH);
digitalWrite(LB22,HIGH);
digitalWrite(LB31,HIGH);
delay(200);
digitalWrite(LB13,LOW);
digitalWrite(LB22,LOW);
digitalWrite(LB31,LOW);
delay(200);
}
//Condição de vitória 1 do jogador 2
if ((mP11==2)&&(mP12==2)&&(mP13==2)){
winP2();
digitalWrite(LP11,HIGH);
digitalWrite(LP12,HIGH);
digitalWrite(LP13,HIGH);
delay(200);
digitalWrite(LP11,LOW);
digitalWrite(LP12,LOW);
digitalWrite(LP13,LOW);
delay(200);
}
//Condição de vitória 2 do jogador 2
if ((mP21==2)&&(mP22==2)&&(mP23==2)){
winP2();
digitalWrite(LP21,HIGH);
digitalWrite(LP22,HIGH);
digitalWrite(LP23,HIGH);
delay(200);
digitalWrite(LP21,LOW);
digitalWrite(LP22,LOW);
digitalWrite(LP23,LOW);
delay(200);
}
//Condição de vitória 3 do jogador 2

```

```

    if ((mP31==2)&&(mP32==2)&&(mP33==2)){
    winP2();
    digitalWrite(LP31,HIGH);
    digitalWrite(LP32,HIGH);
    digitalWrite(LP33,HIGH);
    delay(200);
    digitalWrite(LP31,LOW);
    digitalWrite(LP32,LOW);
    digitalWrite(LP33,LOW);
    delay(200);
    }
    //Condição de vitória 4 do jogador 2
    if ((mP11==2)&&(mP21==2)&&(mP31==2)){
    winP2();
    digitalWrite(LP11,HIGH);
    digitalWrite(LP21,HIGH);
    digitalWrite(LP31,HIGH);
    delay(200);
    digitalWrite(LP11,LOW);
    digitalWrite(LP21,LOW);
    digitalWrite(LP31,LOW);
    delay(200);
    }
    //Condição de vitória 5 do jogador 2
    if ((mP12==2)&&(mP22==2)&&(mP32==2)){
    winP2();
    digitalWrite(LP12,HIGH);
    digitalWrite(LP22,HIGH);
    digitalWrite(LP32,HIGH);
    delay(200);
    digitalWrite(LP12,LOW);
    digitalWrite(LP22,LOW);
    digitalWrite(LP32,LOW);
    delay(200);

```

```

}
//Condição de vitória 6 do jogador 2
if ((mP13==2)&&(mP23==2)&&(mP33==2)){
winP2();
digitalWrite(LP13,HIGH);
digitalWrite(LP23,HIGH);
digitalWrite(LP33,HIGH);
delay(200);
digitalWrite(LP13,LOW);
digitalWrite(LP23,LOW);
digitalWrite(LP33,LOW);
delay(200);
}
//Condição de vitória 7 do jogador 2
if ((mP11==2)&&(mP22==2)&&(mP33==2)){
winP2();
digitalWrite(LP11,HIGH);
digitalWrite(LP22,HIGH);
digitalWrite(LP33,HIGH);
delay(200);
digitalWrite(LP11,LOW);
digitalWrite(LP22,LOW);
digitalWrite(LP33,LOW);
delay(200);
}
//Condição de vitória 8 do jogador 2
if ((mP13==2)&&(mP22==2)&&(mP31==2)){
winP2();
digitalWrite(LP13,HIGH);
digitalWrite(LP22,HIGH);
digitalWrite(LP31,HIGH);
delay(200);
digitalWrite(LP13,LOW);
digitalWrite(LP22,LOW);

```

```
digitalWrite(LP31,LOW);  
delay(200);  
}  
  
//fecha void loop  
//fim de programa
```



```
void piscapiscanatalino(){  
  
    Serial.println("PISCANDO OOOOOO!!!!!!!!!!!!!!!!!!!!!!");  
  
    digitalWrite(LP1,HIGH);  
    digitalWrite(LP2,HIGH);  
    digitalWrite(rele1,LOW);  
    digitalWrite(rele2,LOW);  
}  
void erro(){  
  
Serial.println("erroooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo");  
  
    digitalWrite(LP1,HIGH);  
    digitalWrite(LP2,HIGH);  
    delay(200);  
    digitalWrite(LP1,LOW);  
    digitalWrite(LP2,LOW);  
    delay(200);  
}  
void vezP1(){  
  
    Serial.println("Vez do P1?????????????????????????");  
  
    digitalWrite(rele1,HIGH);  

```

```

    digitalWrite(rele2,LOW);
    digitalWrite(LP1,HIGH);
    digitalWrite(LP2,LOW);

}

void vezP2(){

    Serial.println("Vez do P2*****");

    digitalWrite(rele2,HIGH);
    digitalWrite(rele1,LOW);
    digitalWrite(LP2,HIGH);
    digitalWrite(LP1,LOW);

}


void winP1(){
    Serial.println("Vencedor Player 1");
    digitalWrite(releG,HIGH);
    digitalWrite(LB11,HIGH);//FAZER UM X NO OUTRO JOGADOR PRA
INDICAR DERROTA
    digitalWrite(LB13,HIGH);//
    digitalWrite(LB22,HIGH);//
    digitalWrite(LB31,HIGH);//
    digitalWrite(LB33,HIGH);//
    digitalWrite(LP1,HIGH);// PISCAR O PLAYER QUE VENCEU
    delay(200);
    digitalWrite(LP1,LOW);
    digitalWrite(LB11,LOW);
    digitalWrite(LB13,LOW);
    digitalWrite(LB22,LOW);
    digitalWrite(LB31,LOW);

```

```

    digitalWrite(LB33,LOW);
    delay(200);
}

void winP2(){
    Serial.println("Vencedor Player 2");
    digitalWrite(releG,HIGH);
    digitalWrite(LP11,HIGH);
    digitalWrite(LP13,HIGH);
    digitalWrite(LP22,HIGH);
    digitalWrite(LP31,HIGH);
    digitalWrite(LP33,HIGH);
    digitalWrite(LP2,HIGH);
    delay(200);
    digitalWrite(LP2,LOW);
    digitalWrite(LP11,LOW);
    digitalWrite(LP13,LOW);
    digitalWrite(LP22,LOW);
    digitalWrite(LP31,LOW);
    digitalWrite(LP33,LOW);
    delay(200);
}

```

3.7 – Produção das peças

Optamos por reutilizar materiais e aproveitamos barras de ferro descartadas na escola para fazer as peças. Tiramos as medidas de cada casa do tabuleiro e fizemos as peças com as mesmas medidas para que exista proporção. Cortamos as mesmas com uma serra mecânica presente na oficina da escola, em seguida esmerilhamos e pintamos.



Figura 56 – Peças do Tic Tac Toe

4 – Resultados Obtidos

Após dois anos de trabalho alcançamos o objetivo e transformamos uma grua em um jogo da velha. Com dois pontos de controle, se tornou possível duas pessoas se enfrentarem utilizando um joystick para movimentar o eletroímã que pesca a peça desejada e a coloca numa determinada casa.

O joystick movimenta o eletroímã por uma ponte rolante conectada a dois motores que auxiliam no deslocamento X (carro menor da ponte) e Y (carro maior da ponte). Quando o jogador posiciona o eletroímã perto da peça, ele deverá acionar uma chave para magnetiza-lo. Para soltar a peça ao final do transporte, a chave devera ser desacionada. Para facilitar a aproximação da peça, o jogador também conta com dois botões que sobem e descem o eletroímã – trabalham em Z. Após ter jogado, o jogador deverá acionar uma chave para passar a vez para o adversário jogar. Enquanto isso, dois LEDs vão indicar quando é ou não a vez do adversário. Os jogadores não podem operar os painéis ao mesmo tempo. No painel, o jogador poderá acompanhar suas peças, pois haverão 9 LEDs que indicarão a localização delas no tabuleiro. Em caso de vitória, automaticamente os LEDs irão piscar para indicar que o jogador venceu. Com o término da partida é possível remover as peças do tabuleiro e iniciar uma nova partida.

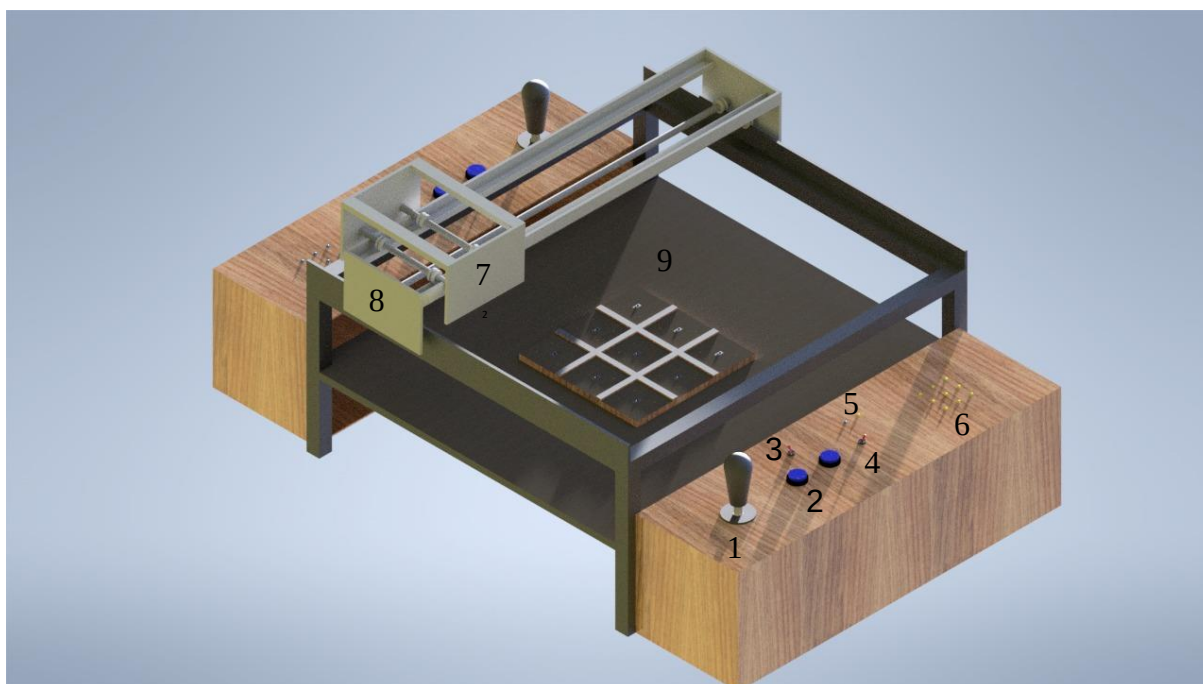


Figura 57 – Tic Tac Toe

- 1 – Joystick
- 2 – Botões sobe/desce
- 3 – Chave de turno
- 4 – Chave eletroímã
- 5 – LEDs indicadores de turno
- 6 – LEDs de posição
- 7 – Carro menor
- 8 – Carro maior
- 9 - Tabuleiro

5 - Conclusão

O projeto foi muito amplo em seu desenvolvimento e foram encontrados alguns obstáculos em seu funcionamento e montagem. Dentro do circuito de potência foram encontrados problemas na soldagem das placas, pois as mesmas foram reutilizadas de projetos passados, sendo necessária o remodelamento da solda para o correto funcionamento. O *joystick* apresentou mau funcionamento em seus fins de curso. A substituição dos mesmos foi necessária. A velocidade do carrinho era muito lenta, para solucionar o problema, fizemos dois eixos que giram com a ajuda de um motor a roda do carro, fazendo o mesmo andar como num trilho.

O trabalho foi fundamental para a absorção dos conhecimentos em mecatrônica. Dentre eles podemos apresentar como principais: aplicação de lógicas de programação para o controle das ações do projeto (passar a vez durante o jogo, indicador de vez, intertravamento, etc); absorção em conhecimentos de mecânica e manufatura que ajudaram a construir a estrutura do projeto e garantir o acabamento do mesmo; tivemos a oportunidade de exercer conhecimentos em comandos e instalações elétricas aplicando-os aos circuitos do projeto como o funcionamento dos botões, joysticks, ímãs, LEDs e integração software-hardware.

O projeto promete trazer bastante entretenimento e atração tecnológica, que são os dois principais pontos para cumprir o objetivo de socialização e conhecimento didático.

Referências

Aramax. Qual é a diferença entre o Aço Galvanizado e o Inoxidável?. Santa Catarina. Disponível em: <<https://aramax.com.br/qual-a-diferenca-entre-aco-galvanizado-e-inoxidavel/>>. Acesso em: 4 de Nov. de 2019.

BERTULUCCI, Cristiano. **Sensores Ópticos: Como Funcionam?** Citisystems. Disponível em:<<https://tecnoblog.net/247956/referencia-site-abnt-artigos/>>. Acesso em: 4 de Nov. de 2019.

Engenharia de Movimentação – CSA. Santa Catarina. Disponível em: <<https://www.csm.ind.br/engenharia/produto/pontes-rolantes-movimentacao-icamento-de-cargas/>>. Acesso em: 4 Nov. 2019.

FILIPEFLOP. **Placa MEGA 2560 R3 + Cabo USB para Arduino.** Disponível em:<<https://www.filipeflop.com/produto/placa-mega-2560-r3-cabo-usb-para-arduino/>>. Acesso em: 11 Nov. 2019.