



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Cubinho: Artefato Robótico Motivacional para Pessoas com Transtorno do Espectro Autista Utilizando o Aplicativo Prevenir

João Anselmo Bandeira dos Reis

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Orientador

Prof. Dr. Wilson Henrique Veneziano

Coorientadora

Prof.a M. Sc. Maraísa Helena Borges Estevão Pereira

Brasília
2021

Dedicatória

Dedico esse trabalho aos familiares e amigos que contribuíram de alguma forma para minha formação.

Agradecimentos

Agradeço ao professor Dr. Wilson Henrique Veneziano e à professora M. Sc. Maraísa Helena Borges Estevão Pereira pela dedicação e orientação durante o desenvolvimento desse projeto. Agradeço aos amigos Clara, Eduardo e Matheus pelo companheirismo durante toda essa jornada acadêmica. Agradeço aos meus pais e irmã que contribuíram de incontáveis formas. Agradeço a minha mãe por sempre me indicar o caminho.

Resumo

Neste trabalho será utilizada uma placa com microcontrolador - placa Arduino -, juntamente de outros componentes eletrônicos, para formar um artefato robótico que se conectará ao aplicativo para *smartphones* Prevenir [1], voltado para a prevenção de acidentes domésticos de pessoas com Transtorno do Espectro Autista. Cores, luzes, sons, movimentos e imagens são utilizados para animar o artefato robótico: um cubo de papel panamá personalizável de acordo com as preferências e as necessidades individuais com a finalidade de ser um reforço positivo utilizado como recurso motivacional para reforçar o comportamento esperado - reconhecimento das situações perigosas apresentadas no aplicativo -, uma vez que pessoas no espectro autista podem não ser sensíveis ao reforço social ou a recompensas verbais.

Palavras-chave: Arduino, Prevenção de Acidentes Domésticos, Autismo, Transtorno do Espectro Autista

Abstract

This project uses an Arduino Board and other electronic peripherals to create a robot that can be connected to the smartphone app Prevenir, app that aims to teach people on the Autism Spectrum Disorder (ASD) how to recognize and avoid dangerous situations at home. Colors, lights, sounds, movement and images are used to give personality to the robot: a cube made of a strong type of paper that can be customized to fit the needs and preferences of each individual aiming to motivate and reward the expected behavior - acknowledge dangerous situations inside the app - since regular motivations such as social incentives may be inefficient to people in the autism spectrum.

Keywords: Arduino, Domestic Accidents Prevention, Autism, Autism Spectrum Disorder

Sumário

1	Introdução	1
1.1	Problema	1
1.2	Justificativa	2
1.3	Objetivo Geral	2
1.4	Objetivos Específicos	2
1.5	Procedimentos Adotados	3
1.6	Organização desse Trabalho	4
2	Transtorno do Espectro Autista	5
2.1	Introdução	5
2.2	Características, diagnósticos e causas	6
2.3	Intervenções	7
2.4	Legislação	9
2.5	Tecnologia Assistiva	10
2.6	Acessibilidade na Web	12
3	O Arduino	14
3.1	O <i>Software</i>	14
3.1.1	A IDE	15
3.1.2	Linguagem de Programação do Arduino	17
3.2	O Hardware	18
3.2.1	Microcontrolador	19
3.2.2	Entradas e Saídas Digitais	19
3.2.3	Saídas PWM	20
3.2.4	Entradas Analógicas	21
3.3	Por que o Arduino?	22
4	Componentes Eletrônicos	23
4.1	Arduinos	23
4.1.1	Arduino Mega	24

4.1.2	Arduíno Uno	24
4.2	Ponte H e Motor de Corrente Contínua	24
4.3	Motor Servo	26
4.4	Módulo <i>Bluetooth</i>	27
4.5	LEDs	28
4.6	Alto Falante	30
4.7	<i>Display</i> TFT	31
5	Testes dos Componentes	33
5.1	Arduínos	33
5.2	Ponte H e Motor de Corrente Contínua	33
5.3	Motor Servo	34
5.4	Módulo Bluetooth	35
5.5	LEDs	37
5.6	Alto Falante	37
5.7	Display TFT	39
6	O Hardware Desenvolvido	42
6.1	Montagem dos Componentes	42
6.1.1	O Cubo	45
6.1.2	A ponte H e os motores DC	45
6.1.3	Os motores Servo	46
6.1.4	O Módulo <i>Bluetooth</i>	46
6.1.5	Os LEDs e o Alto Falante	49
6.1.6	O <i>display</i> TFT	50
6.1.7	Montagem Final	52
6.2	O Cubinho	52
6.2.1	A Carcaça	53
6.2.2	Os Reforçadores	54
6.3	O Aplicativo Prevenir	59
7	Conclusão	64
	Referências	65

Lista de Figuras

3.1	Arduino IDE.	15
3.2	Gerenciador de bibliotecas do Arduino.	16
3.3	Linguagem de programação Arduino.	18
3.4	Função PWM[2].	21
4.1	Arduino Mega 2560 Datasheet [3].	25
4.2	Arduino Mega 2560 Pinout [3].	26
4.3	Arduino Uno Rev3 Datasheet [4].	27
4.4	Arduino Uno Rev3 Pinout [4].	28
4.5	Ponte H Pinout.	29
4.6	Motor DC.	29
4.7	Motor Servo	30
4.8	Módulo Bluetooth.	30
4.9	LED Vermelho	31
4.10	Alto Falante	31
4.11	<i>Display</i> TFT <i>Pinout</i>	32
4.12	<i>Display</i> TFT	32
5.1	Esquemático do circuito de teste dos Arduinos e LED [5]	34
5.2	Código de teste para os Arduinos [5]	34
5.3	Código de teste para a ponte H e os motores DC.	35
5.4	Esquemático do circuito de teste para a ponte h e os motores DC.	36
5.5	Esquemático do circuito de testes do motor servo [6].	36
5.6	Código de teste do motor servo	37
5.7	Esquemático do circuito de teste para o módulo bluetooth.	38
5.8	Código de teste para o módulo bluetooth.	38
5.9	Código de teste para os LEDs.	39
5.10	Código de teste para o Alto Falante.	39
5.11	Esquemático do circuito de teste para o Alto Falante.	40

5.12	Código de teste para o o display TFT.	41
6.1	Esquemático em blocos do Hardware.	43
6.2	Esquemático da montagem dos componentes.	44
6.3	Cubo de papel panamá.	45
6.4	Montagem da ponte H e dos motores DC.	46
6.5	Montagem da ponte H e dos motores DC no cubo.	47
6.6	Montagem do Motor Servo.	47
6.7	Montagem do Módulo Bluetooth.	48
6.8	Montagem do Módulo Bluetooth no Cubo.	48
6.9	sn74hc595n Pinout [7].	49
6.10	Montagem dos LEDs e do Alto Falante.	50
6.11	Montagem dos LEDs e alto falante no Cubo.	51
6.12	Montagem do <i>Display</i> TFT no Cubo.	51
6.13	Montagem final do Cubinho.	52
6.14	Cubinho sem a parte de cima.	53
6.15	Cubinho montado.	53
6.16	Imagem utilizada no rosto do Cubinho.	54
6.17	Cubinho personalizado 1.	55
6.18	Cubinho personalizado 2.	55
6.19	Diagrama de blocos do Código do Arduino.	56
6.20	Parte de baixo do Cubinho.	57
6.21	Mão com polegar do Cubinho.	58
6.22	Mão com indicador do Cubinho.	58
6.23	Parte de cima do Cubinho com os LEDs.	59
6.24	Alto falante grudado no Cubinho.	60
6.25	<i>Display</i> TFT no Cubinho.	60
6.26	Prevenir Atividade 1.	61
6.27	Prevenir Atividade 2.	61
6.28	Tela de configurações do aplicativo Prevenir.	62
6.29	Tela de configuração do Cubinho no aplicativo Prevenir.	62
6.30	Tela de configuração do <i>bluetooth</i> no aplicativo Prevenir.	62
6.31	Tela de configurações dos reforços no aplicativo Prevenir.	63
6.32	Tela de seleção dos reforços no aplicativo Prevenir.	63

Capítulo 1

Introdução

A tecnologia móvel em *smartphones* e *tablets* está cada vez mais presente na vida das pessoas, incluindo aquelas que possuem alguma necessidade especial. Indivíduos com Transtorno do Espectro Autista (TEA) podem apresentar dificuldades para aprender habilidades requeridas na vida diária [8] e os dispositivos tecnológicos podem auxiliar esse aprendizado e ajudar na aquisição de habilidades.

De acordo com Caminha et. al (2016) [9] a tecnologia pode ser atrativa para as pessoas do espectro autista, despertando o interesse e a atenção e contribuindo para a aprendizagem. Brok e Sterkenburg (2015) [10], por sua vez, em uma revisão sistemática, relataram que algumas tecnologias podem ser usadas para aprender tipos específicos de habilidades - vídeos em computadores ou dispositivos portáteis para habilidades de vida diária, realidade virtual para percepção do tempo e emoções. Ressaltaram que, para a aquisição de conceitos cognitivos que estão sendo vistos pela primeira vez, é conveniente usar tecnologias mais atuais por apresentar mais recursos para suportar a aprendizagem.

Por isso, entende-se que a utilização de uma placa Arduíno, juntamente de periféricos eletrônicos, que utiliza cores, luzes, sons, movimento e imagens, dentro de um cubo personalizável para reforçar um comportamento, seja o complemento necessário para apoiar o uso de um aplicativo móvel destinado a prevenir acidentes domésticos para pessoa com TEA. É importante ressaltar que os elementos reforçadores acima descritos podem ser adaptados e customizados de acordo com a preferência individual, uma vez que nem todas as pessoas com TEA irão se interessar pelo mesmo reforço.

1.1 Problema

Pode ser bastante difícil para uma pessoa com transtorno do espectro autista prestar atenção e aprender da mesma maneira que uma criança com desenvolvimento típico [8]. Para estas, elogios e aprovação são reforçadores suficientemente eficazes. Contudo, para

pessoas com TEA esse tipo de reforço pode não funcionar adequadamente, advindo a necessidade de reforçadores singulares, como luzes, cores, sons, movimento e imagens adaptáveis à preferência individual, capazes de causar a repetição do comportamento desejado. [11]

1.2 Justificativa

Para auxiliar pais e responsáveis e tendo como foco as dificuldades encontradas por pessoas com TEA, foi desenvolvido o aplicativo Prevenir, voltado para a prevenção de acidentes domésticos. Nessa época em que a pandemia em razão do novo coronavírus obrigou a todos a passar mais tempo em casa, essa iniciativa é ainda mais importante, uma vez que o ambiente doméstico é uma grande fonte de internações hospitalares e de mortes de crianças [12] e pode ser bastante difícil para pessoas com TEA reconhecer e evitar situações de risco mesmo no âmbito doméstico.

Muitos indivíduos com TEA têm déficits em sua capacidade de realizar algumas atividades independentemente e precisam de instrução regular para dominar as habilidades da vida diária [8]. Além disso, os indivíduos com TEA podem ter dificuldade para aprender pela observação do outro, necessitando que cada habilidade ou comportamento seja trabalhado metodicamente, uma vez que reforçadores comuns eficazes para crianças com desenvolvimento típico podem não trazer resultados para crianças com autismo, sendo necessário usar reforçadores pessoais, os quais só fazem sentido para aquele indivíduo [13].

Assim, tendo em vista que pessoas com TEA podem necessitar de reforçadores diferenciados para que fiquem motivadas a aprender [11], esse dispositivo eletrônico (Cubinho), integrado ao software de prevenção de acidentes domésticos (Prevenir) traz reforçadores customizáveis que objetivam aumentar a chance de um comportamento alvo – reconhecimento da situação de risco - ocorrer novamente.

1.3 Objetivo Geral

O objetivo geral deste trabalho é o desenvolvimento de um artefato robótico em forma de cubo utilizando a placa Arduino conectada a periféricos eletrônicos e sua integração ao software Prevenir, para que ele ofereça uma variedade de reforçadores comportamentais na ocorrência do reconhecimento correto das situações de risco apresentadas no aplicativo.

1.4 Objetivos Específicos

Com o intuito de alcançar o objetivo geral, foram definidas algumas métricas específicas:

- Levantamento de requisitos educacionais junto a uma psicopedagoga especializada em autismo;
- Integração do *display* para exibição das imagens;
- Integração do módulo *bluetooth* para conexão com o aplicativo no celular;
- Integração dos motores para movimentação do Cubo;
- Desenvolvimento do código-fonte para o Arduino;
- Conexão e testes com o aplicativo Prevenir

1.5 Procedimentos Adotados

A confecção desse projeto se deu com a realização de pesquisas bibliográficas em livros, sites e artigos sobre o Transtorno do Espectro Autista e a placa com microcontrolador Arduino e seus periféricos. Professores especialistas em educação especial contribuíram com a elaboração da proposta, dirimindo dúvidas quanto ao processo de aprendizagem das pessoas com TEA. A duração do desenvolvimento desse trabalho foi de aproximadamente um ano e quatro meses e consistiu das seguintes etapas:

- Observação e análise das necessidades encontradas por pessoas com Transtorno do Espectro Autista no ambiente domiciliar.
- Estudo das formas de aprendizado de pessoas com autismo e de como incentivá-las a continuar aprendendo.
- Elaboração do conceito do *hardware* baseado nas informações adquiridas.
- Estudo sobre *displays* que apresentar imagens motivacionais no cubo.
- Estudo para definir o *hardware* de controle do sistema.
- Estudo sobre a comunicação sem fio entre *smartphones* e a placa microcontroladora.
- Desenvolvimento do código fonte para controle do *hardware*.
- Testes funcionais dos componentes eletrônicos individualmente.
- Testes funcionais do código completo que controla o *hardware*.
- Desenvolvimento do código para conexão do *hardware* com o aplicativo Prevenir.
- Testes funcionais da conexão do aplicativo com o cubo.
- Testes funcionais do cubo finalizado.
- Customização do exterior do cubo.

1.6 Organização desse Trabalho

Esta monografia está organizada nos seguintes capítulos:

- Capítulo 2 - Descreve as características principais do autismo, alguns tipos de intervenção, legislação sobre o assunto, como as tecnologias podem afetar pessoas com TEA e fornece um contexto histórico.
- Capítulo 3 - Descreve as principais características do Arduino, desde o *software* até o *hardware*.
- Capítulo 4 - Descreve as principais características e funcionalidades dos componentes eletrônicos utilizados.
- Capítulo 5 - Descreve como ocorreram os testes dos componentes eletrônicos.
- Capítulo 6 - Descreve como ocorreu a montagem do Cubinho e suas funcionalidades.
- Capítulo 7 - Descreve a conclusão desse trabalho, apresentando possíveis melhorias para o projeto.

Capítulo 2

Transtorno do Espectro Autista

Este capítulo descreve diversas informações relevantes sobre o Transtorno do Espectro Autista, desde o histórico de sua descoberta, suas principais características, possíveis causas e tipos e efeitos de intervenções. Essas informações são a base do desenvolvimento desse trabalho e o entendimento das necessidades do público foi o fator de maior relevância para todas as tomadas de decisões.

2.1 Introdução

Autismo deriva do grego e significa “autos”, voltar-se para si mesmo. Eugen Bleuler, psiquiatra que estudava a esquizofrenia, foi quem usou a palavra autismo pela primeira vez em 1911. [14]

O médico austríaco Leo Kanner, em 1943, em seu artigo *Autistic Disturbances of Affective Contact* (Distúrbios Autísticos do Contato Afetivo) [15], descreveu 11 (onze) casos de crianças que apresentavam com certa frequência comportamentos estereotipados ou repetitivos, além de dificuldades na comunicação e no contato social e afetivo.

Paralelamente, em 1944, Hans Asperger, médico também austríaco, elaborou um artigo intitulado “A psicopatia autista na infância” [16] que não teve muita repercussão na época, uma vez que foi publicado em alemão e durante a Segunda Guerra Mundial. Assim, apenas na década de 1980 foi reconhecido como um dos pioneiros no estudo do autismo, tendo seu nome associado à síndrome que identificou - Síndrome de Asperger. [14]

Em 1979, Lorna Wing e Judith Gould relataram em *Severe impairments of social interaction and associated abnormalities in children: Epidemiology and classification* (Prejuízos graves de interação social e anormalidades associadas em crianças: Epidemiologia e classificação) [17] déficits específicos em três áreas – domínio social, domínio do pensamento e do comportamento e domínio da linguagem e da comunicação - e propuseram uma tríade de incapacidades, a Tríade de Wing. Conforme a autora, os elementos da sua tríade são

variáveis e flexíveis, podendo aparecer em infinitas combinações. Wing também propôs o conceito do que atualmente é conhecido por espectro.

A partir da década de 1980, os manuais psiquiátricos retiraram o autismo da categoria de psicose, passando a fazer parte dos Transtornos Globais do Desenvolvimento (CID 10 - Classificação Estatística Internacional de Doenças, 2004) e dos Transtornos Invasivos do Desenvolvimento do Manual Diagnóstico e Estatístico de Transtornos Mentais – DSM-IV. [14]

Atualmente está em vigor a 5ª edição do Manual Diagnóstico e Estatístico de Transtornos Mentais (DSM-5) [18], publicado em 2013, bem como a recém-publicada 11ª versão da Classificação Estatística Internacional de Doenças e Problemas Relacionados com a Saúde (CID-11) [19], elaborada pela Organização Mundial de Saúde (OMS), que estabelece, de modo geral, critérios diagnósticos do TEA semelhantes ao DSM-V.

2.2 Características, diagnósticos e causas

Conforme a última versão do Manual de Diagnóstico e Estatística de Doenças Mentais da Academia Americana de Psiquiatria - DSM V de 2013 [18], o autismo é um transtorno do neurodesenvolvimento, caracterizado por dificuldades de comunicação social e de interação social, bem como pela presença de comportamentos, interesses ou atividades repetitivos ou restritos.

Os sintomas trazidos no DSM-5 formam o cerne do autismo, mas a intensidade com que se apresentam é variável. Ainda que os sintomas variem de pessoa para pessoa, esses elementos são determinantes para realizar o diagnóstico de autismo. Por existir uma pluralidade de formas de manifestação do autismo, podendo variar da forma leve a mais severa, as novas classificações passaram a adotar o nome “Transtorno do Espectro Autista”. Em razão dessas gradações na presença e na intensidade dos sintomas, acarretando em diferentes graus de comprometimento, nenhuma pessoa com autismo é igual a outra. Assim, o quadro clínico é diferenciado e individualizado, existindo em torno dos sintomas centrais outros sintomas secundários.

Segundo a Sociedade Brasileira de Pediatria [8] “O TEA tem origem nos primeiros anos de vida, mas sua trajetória inicial não é uniforme. Em algumas crianças, os sintomas são aparentes logo após o nascimento. Na maioria dos casos, no entanto, os sintomas do TEA só são consistentemente identificados entre os 12 e 24 meses de idade”.

Apesar de as características típicas do TEA estarem presentes desde muito cedo, muitas crianças são diagnosticadas tardiamente, a partir dos 4 anos de idade. Ainda em conformidade com Sociedade Brasileira de Pediatria [8], esse diagnóstico tardio é prejudi-

cial, uma vez que quanto mais cedo o autismo for diagnosticado e a intervenção for feita mais ganhos a criança terá na cognição e no funcionamento adaptativo.

Quanto às causas, a Associação Brasileira de Pediatria diz que o TEA pode se originar de uma combinação genética com fatores ambientais. Mello (2007) em seu *Autismo: guia prático* [20], corrobora esse entendimento:

“As causas do autismo são desconhecidas. Acredita-se que a origem do autismo esteja em anormalidades em alguma parte do cérebro ainda não definida de forma conclusiva e, provavelmente, de origem genética. Além disso, admite-se que possa ser causado por problemas relacionados a fatos ocorridos durante a gestação ou no momento do parto.

A hipótese de uma origem relacionada à frieza ou rejeição materna já foi descartada, relegada à categoria de mito há décadas. Porém, a despeito de todos os indícios e da retratação pública dos primeiros defensores desta teoria, persistem adeptos desta corrente que ainda a defendem ou defendem teorias aparentemente diferentes, mas derivadas desta.”

Mello (2007) [20] salienta que pode ser recomendado para prevenir o autismo os cuidados gerais que toda gestante deve ter, tais como cuidados com ingestão de produtos químicos, remédios, álcool ou fumo.

De acordo com a Associação de Amigos do Autista [21], apesar de não existirem marcadores biológicos e exames específicos para autismo, alguns exames, tais “como o cariótipo com pesquisa de X frágil, o eletroencefalograma (EEG), a ressonância magnética nuclear (RNM), os erros inatos do metabolismo, o teste do pezinho, as sorologias para sífilis, rubéola e toxoplasmose; a audiometria e testes neuropsicológicos podem ser necessários para investigar as causas e doenças associadas.”

2.3 Intervenções

A intervenção precoce já na primeira infância é primordial para o desenvolvimento das pessoas com TEA. Apesar de não haver cura para o TEA, essas intervenções podem auxiliar a superar as dificuldades enfrentadas e impactar positivamente no que diz respeito a sua autonomia e qualidade de vida.

Muitas pessoas com TEA podem viver de forma independente, entretanto, existem outras com dificuldades mais severas que precisarão de cuidados e atenção mais prolongados e podem ter uma capacidade limitada para as atividades rotineiras do dia a dia e para participarem da sociedade.

O trabalho com pessoas com TEA pode ser feito a partir de diversas abordagens. Qualquer que seja a intervenção adotada, as terapias devem focar nas particularidades

e nas necessidades de cada indivíduo. Os tratamentos visam favorecer a autonomia e, assim, propiciar qualidade de vida à pessoa com TEA e à sua família.

De acordo com Mello (2007) [20], os métodos e programas educacionais mais usuais para o autismo são o TEACCH, ABA e o PECS.

O TEACCH (*Treatment and Education of Autistic and related Communication handicapped Children*) - Tratamento e Educação para Crianças Autistas e com Distúrbios Correlatos da Comunicação – foi criado em 1966 na Carolina do Norte(EUA). Explica Mello (2007) [20]:

“O método TEACCH utiliza uma avaliação chamada PEP-R (Perfil Psicoeducacional Revisado) para avaliar a criança, levando em conta os seus pontos fortes e suas maiores dificuldades, tornando possível um programa individualizado.

O TEACCH se baseia na organização do ambiente físico através de rotinas - organizadas em quadros, painéis ou agendas - e sistemas de trabalho, de forma a adaptar o ambiente para tornar mais fácil para a criança compreendê-lo, assim como compreender o que se espera dela. Através da organização do ambiente e das tarefas da criança, o TEACCH visa desenvolver a independência da criança de modo que ela necessite do professor para o aprendizado, mas que possa também passar grande parte de seu tempo ocupando-se de forma independente.”

Já o PECS - *Picture Exchange Communication System* ou Sistema de Comunicação por troca de figuras- auxilia a pessoa com TEA a compreender que com a comunicação é possível obter mais facilmente o que deseja. O método utiliza cartões para ajudar na comunicação de crianças não oralizadas e na organização da linguagem verbal daquelas oralizadas, mas que possuem limitações e dificuldades e necessitam de ajuda para organizar a linguagem. [22]

O ABA - Análise Aplicada ao Comportamento (*Applied Behavior Analysis*) - é na verdade um ramo aplicado da ciência da análise do comportamento, que pode ser utilizada em muitos segmentos da sociedade. A partir da década de 1980, surgiram estudos mostrando como esse tipo de análise poderia auxiliar no tratamento de pessoas com TEA. Baseia-se no condicionamento operante e reforçadores, tendo como foco incentivar determinados comportamentos e reduzir outros, bem como fortalecer habilidades existentes e desenvolver habilidades ainda não despertadas. Há várias técnicas e estratégias de ensino e tratamento comportamentais associados a análise do comportamento aplicada que têm se mostrado útil no contexto da intervenção, incluindo tentativas discretas, análise de tarefas, ensino incidental e análise funcional. [23]

Nesse tipo de intervenção o uso de reforçadores é muito importante. Quando uma criança neurotípica fala as primeiras palavras, os pais reforçam esse comportamento sorrindo ou dando atenção. No caso de uma criança com transtorno do espectro autista, ela pode não atender ao reforço social e precisarão ser usado reforçadores tangíveis, tais como brinquedos, comida, vídeos, etc. [11]

Com o passar do tempo, o reforço deve ser retirado para evitar a dependência da pessoa com TEA e usar o reforçamento natural com recompensas verbais.

Explica Mello [20]:

“O tratamento comportamental analítico do autismo visa ensinar à criança habilidades que ela não possui, através da introdução destas habilidades por etapas. Cada habilidade é ensinada, em geral, em esquema individual, inicialmente apresentando-a associada a uma indicação ou instrução. Quando necessário, é oferecido algum apoio (como por exemplo, apoio físico), que deverá ser retirado tão logo seja possível, para não tornar a criança dependente dele. A resposta adequada da criança tem como consequência a ocorrência de algo agradável para ela, o que na prática é uma recompensa. Quando a recompensa é utilizada de forma consistente, a criança tende a repetir a mesma resposta.”

2.4 Legislação

A Constituição Federal de 1988 [24], as Constituições Estaduais, os Tratados e Convenções Internacionais, as Leis Federais, Estaduais, Distritais e Municipais e outros diplomas normativos asseguram diversos direitos à pessoa com deficiência. Especificamente para a pessoa com TEA existe a Lei Berenice Piana – Lei 12.764, de 27 de dezembro de 2012 [25] - que instituiu a Política Nacional de Proteção dos Direitos da Pessoa com Transtorno do Espectro Autista. Esta Lei estipulou que a pessoa com TEA é considerada pessoa com deficiência para todos os efeitos legais. Isso permitiu que estes indivíduos fossem abrangidos pelas leis que tratam especialmente de pessoas com deficiência, assim como pelas normas internacionais nas quais o Brasil é signatário, como a Convenção das Nações Unidas sobre os Direitos das Pessoas com Deficiência (Decreto Federal 6.949/2000 [26]).

A Lei nº 13.146 de 2015 [27], por sua vez, instituiu a Lei Brasileira de Inclusão da Pessoa com Deficiência (Estatuto da Pessoa com Deficiência) e definiu quem é a pessoa com deficiência, determinando como deve ser a sua reabilitação, bem como a obrigatoriedade no diagnóstico e intervenção precoces.

Vale ainda mencionar o Estatuto da Criança e do Adolescente, que garantiu o direito ao respeito da dignidade da criança, bem como a inviolabilidade da integridade física, psíquica e moral; bem como, o Estatuto do Idoso (Lei 10.741/2003 [28]), para os maiores de 60 anos; e a alteração ocorrida na LDB (Lei 9.394/1996) - Lei de Diretrizes e Bases da Educação Nacional - em 2018, que determinou como dever do Estado garantir que a educação especial para crianças entre 0 e 6 anos se estenda ao longo da vida para as pessoas com deficiência [29].

As pessoas com TEA se inserem em outras previsões legais para a garantia de outros direitos, sendo que, sem esgotar os assuntos, podem ser citados os seguintes:

- Direitos referentes à assistência social, assegurados pela Lei Orgânica da Assistência Social (LOAS - Lei 8.742/93 [30]) que disponibiliza benefícios para pessoas com deficiência;
- Direito à educação previsto no Estatuto da Criança e Adolescente, na Lei de Diretrizes e Bases da Educação Nacional (Lei Federal 9.394/96 [31]) e no Decreto 7611/2011 [32]
- Apoio às pessoas com deficiência e sua integração social, tutela jurisdicional de interesses coletivos ou difusos dessas pessoas, atuação do Ministério Público, definição de crimes, outras providências, com apoio na Lei Federal 7.853/89 [33].
- Direito ao transporte com a concessão de passe livre para as pessoas portadoras de deficiência nos termos da Lei Federal 8.899/94 [34].
- Prioridade no atendimento assegurado na Lei Federal 10.048/2000 [35].

2.5 Tecnologia Assistiva

Tecnologia Assistiva, segundo o Comitê de Ajudas Técnicas - CAT, criado pela Portaria 142, de 16 de novembro de 2006[36], "(...) é uma área do conhecimento, de característica interdisciplinar, que engloba produtos, recursos, metodologias, estratégias, práticas e serviços que objetivam promover a funcionalidade, relacionada à atividade e participação de pessoas com deficiência, incapacidades ou mobilidade reduzida, visando sua autonomia, independência, qualidade de vida e inclusão social".

Já a Lei nº 13.146/2015, Lei Brasileira de Inclusão da Pessoa com Deficiência[27], em seu artigo 3º, inciso III, considera tecnologia assistiva como:

“produtos, equipamentos, dispositivos, recursos, metodologias, estratégias, práticas e serviços que objetivem promover a funcionalidade, relacionada à atividade e à participação da pessoa com deficiência ou com mobilidade reduzida, visando à sua autonomia, independência, qualidade de vida e inclusão social”.

Apesar de ser um termo relativamente novo, os recursos e apoios sempre foram utilizados. Podem ser recursos simples como bengalas até produtos que usam tecnologia de ponta. Os recursos de tecnologia assistiva são organizados ou classificados levando em conta as suas funções. A classificação a seguir foi feita em 1998 por José Tonolli e Rita Bersch e foi atualizada para corresponder aos avanços que ocorreram nas respectivas áreas [37].

- Auxílios para a vida diária são materiais e produtos que contribuem para a execução autônoma e independente das tarefas do dia a dia, tais como comer, cozinhar, vestir-

se. Podem ser citados como exemplo o velcro, argolas para zíper, engrossadores de lápis, barras de apoio.

- Comunicação Aumentativa e Alternativa são recursos, eletrônicos ou não, que propiciam o processo comunicativo para pessoas com dificuldades de fala, escrita ou compreensão. São muito utilizadas as pranchas de comunicação com os símbolos PCS ou Bliss, além de vocalizadores e softwares dedicados para este fim
- Recursos de acessibilidade são hardware e software planejados para fazer o computador exequível para pessoas com deficiências. Podem ser mouses, teclados especiais, softwares de reconhecimento de voz, leitores de tela braille ou com sintetização de voz, entre outros.
- Sistemas de controle de ambiente são sistemas eletrônicos que possibilitam que pessoas com limitação locomotora possam gerenciar de forma remota aparelhos eletroeletrônicos, tais como sistemas de segurança, sistemas de climatização, televisores e muito mais.
- Projetos arquitetônicos para acessibilidade são todos os tipos de adaptações estruturais feitas em um ambiente para reduzir barreiras físicas e facilitar o acesso e a locomoção. São rampas, elevadores, adaptações em banheiros.
- Próteses são peças artificiais que substituem membros ou partes ausentes do corpo. Órteses são dispositivos posicionados junto ao corpo e que auxiliam na mobilidade de pessoas com deficiência ou limitação física.
- Adequação postural são adaptações que objetivam proporcionar posturas confortáveis, estáveis e alinhadas. Nessa categoria estão almofadas, assentos e encostos anatômicos.
- Auxílios de mobilidade são os equipamentos que ajudam na mobilidade, tais como bengalas, andadores, muletas, cadeiras de rodas e outros.
- Auxílios para ampliação da função visual e recursos que traduzem conteúdos visuais em áudio ou informação tátil são equipamentos que possibilitam a aquisição de informações para pessoas que necessitem de auxílio com a visão. Como exemplo pode-se citar lupas, lentes, sintetizadores de voz, ampliadores de tela, materiais com texturas e relevos.
- Auxílios para melhorar a função auditiva e recursos utilizados para traduzir os conteúdos de áudio em imagens, texto e língua de sinais são quaisquer aparelho ou dispositivo que viabilize a assimilação de pessoas com problemas na audição. Cita-

se, como exemplos, aparelhos para surdez, sistemas com alerta táctil-visual, textos digitais em línguas de sinais entre outros.

- Mobilidade em veículos são utensílios que permitem que uma pessoa com deficiência possa dirigir um carro, que facilitam o embarque e o desembarque. Traduzem-se em elevadores e rampas para cadeira de rodas.
- Esporte e Lazer é tudo que facilita a participação da pessoa com deficiência em atividades esportivas e em atividades de lazer.

2.6 Acessibilidade na Web

A W3C (World Wide Web Consortium) é uma organização internacional que desenvolve especificações técnicas e orientações para web.

A Cartilha de Acessibilidade na Web disponibilizada pela W3C Brasil [38] diz que:

“Acessibilidade na web é a possibilidade e a condição de alcance, percepção, entendimento e interação para a utilização, a participação e a contribuição, em igualdade de oportunidades, com segurança e autonomia, em sítios e serviços disponíveis na web, por qualquer indivíduo, independentemente de sua capacidade motora, visual, auditiva, intelectual, cultural ou social, a qualquer momento, em qualquer local e em qualquer ambiente físico ou computacional e a partir de qualquer dispositivo de acesso.”

A acessibilidade está prevista no WCAG (Web Content Accessibility Guidelines), que traz os padrões de acessibilidade para a web visando promover a eliminação de barreiras e estimular a construção de páginas de forma a serem acessadas por todos. Esse material é referência no mercado e no ambiente acadêmico sobre recomendação de acessibilidade web.

Assim, em conformidade com as Cartilhas da W3C Brasil [38], para um site ser acessível deve seguir os seguintes princípios:

- Perceptível: as informações e a interface são apresentadas de uma forma que possam ser percebidas;
- Operável: a Interface e a navegação devem ser operáveis para todos os usuários;
- Compreensível: a informação deve ser apresentada de forma simples e compreensível;
- Robusto: o conteúdo deve ser robusto de uma forma que possa maximizar sua compatibilidade com diferentes tipos de pessoas e tecnologias assistivas.

O governo brasileiro mantém o Modelo de Acessibilidade em Governo Eletrônico (eMAG)[39], uma versão especializada do WCAG, que visa permitir que “a implementação da acessibilidade digital seja conduzida de forma padronizada, de fácil implementação,

coerente com as necessidades brasileiras e em conformidade com os padrões internacionais.”

Contudo, seguir as recomendações gerais podem não ser suficientes para assegurar a acessibilidade. No que tange especificamente as pessoas com TEA, Britto (2016)[40] afirma que “Uma interação e experiência de uso projetados inadequadamente para pessoas com TEA podem aumentar o esforço de utilização, causar estresse, irritação e ansiedade desnecessários. Além disso, uma solução computacional inacessível pode impedir que os objetivos pedagógicos e terapêuticos sejam alcançados devido às barreiras de interação.”

Britto [41], em seu Guia de Acessibilidade de Interfaces Web com foco em aspectos do autismo, define recomendações que devem ser seguidas para que uma interface web seja acessível para pessoas com TEA. Como recomendações relevantes, pode-se citar permitir a customização visual, eliminar distrações e possuir uma interface minimalista.

Complementando as informações acima, no artigo Recomendações para o Desenvolvimento de Softwares Voltados para Crianças com Transtorno do Espectro Autista, as autoras Heloise Cristini Magaton e Silvia Amélia Bim[42] trazem observações, dentre as quais:

- Sons e imagens condizentes com a realidade: Imagens condizentes com a realidade, de maneira que as crianças TEA consigam reconhecê-las e associá-las ao seu dia-dia. Deve-se também tomar cuidado com os sons, para que não haja repetições errôneas por parte das crianças, (...)
- Uso de recursos visuais/sonoros para incentivos: Durante a interação dos participantes TEA desta pesquisa com os aplicativos escolhidos para a pesquisa de campo, notou-se que ao interagir com aplicativos que possuíam algum tipo de incentivo ao realizar uma ação correta foi reforçador, levando os participantes a continuarem a atividade. Porém, o que também foi observado é que a inclusão de recursos de recompensas e incentivos deve ser feita com parcimônia. (...)
- Atenção com recursos sonoros: Nas observações pode-se verificar que a emissão dos sons dos jogos causou reações inesperadas (e acredita-se, não intencionadas pelos(as) designers do jogo). Por exemplo, a preferência por parte de algumas crianças observadas pelo som de indicação de erro sobre o do acerto. As crianças acabavam errando propositalmente (...) para poder escutar o som agradável aos seus ouvidos. Este resultado sugere a necessidade de mais pesquisas sobre a relação de crianças com TEA e tipos de sons.

Capítulo 3

O Arduino

O Arduino é uma plataforma eletrônica *open-source* de *hardware* e *software* que surgiu como um projeto de pesquisa de Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino e David Mellis no Instituto de Design de Interação de Ivrea (Interaction Design Institute Ivrea - IDII) em Turim, na Itália [43]. O objetivo inicial do Arduino, quando foi introduzido em 2005, foi de facilitar, para os estudantes de design, o desenvolvimento de protótipos eletrônicos, com linguagem acessível e forte relação entre o aspecto físico e o virtual. Essa facilidade permitiu que esses alunos, que não possuíam conhecimentos específicos prévios, pudessem se aventurar no mundo da eletrônica [44].

Com o passar do tempo o Arduino foi se popularizando e se tornou a ferramenta de prototipação mais popular entre engenheiros e grandes corporações. Parte desse sucesso se deve ao fato de ser um projeto *open-source*, que possibilita a qualquer pessoa com interesse, contribuir com testes, correções de *bugs*, criação de tutoriais e exemplos e a adição de novas funcionalidades, gerando assim uma comunidade que o mantém relevante, aumentando a vida útil do projeto [43]. O Arduino é composto fundamentalmente por duas partes, o *hardware*, uma placa eletrônica contendo um microcontrolador e diversos pinos para conexão e o *software*, desenvolvido para controlar a placa.

3.1 O *Software*

O *software* engloba tudo que envolve a parte de programação do Arduino, desde o local onde o código é escrito até a forma de escrevê-lo. É subdividido em duas partes principais, a IDE (*The Arduino Software - IDE*) e a linguagem de programação Arduino (*Arduino Programming Language*).

3.1.1 A IDE

A IDE é um ambiente integrado de desenvolvimento que se comunica com a placa Arduino podendo carregar programas, trocar informações, mostrar resultados, entre outras funcionalidades que auxiliam no desenvolvimento do programa. A IDE pode ser visualizada na Figura 3.1, ela possui um editor de texto para escrever o código, uma área de mensagens, um console e uma barra de ferramentas (*toolbar*) com botões para os menus e as principais funcionalidades da IDE [45].

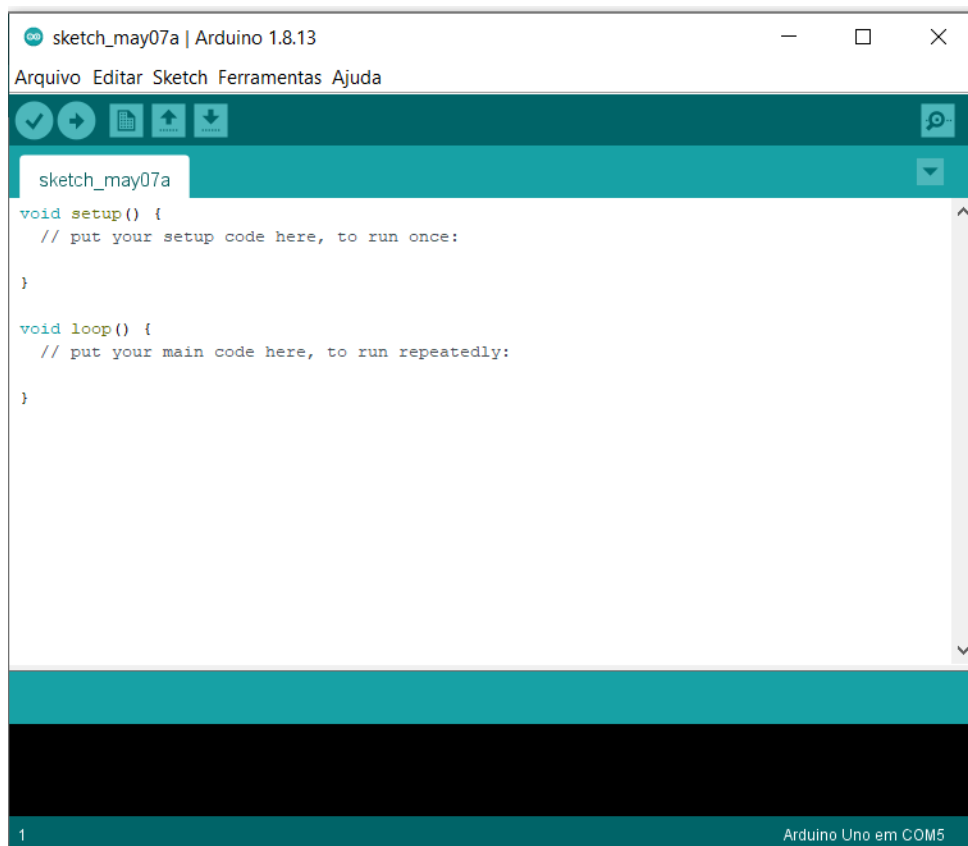


Figura 3.1: Arduino IDE.

A base do ambiente de desenvolvimento e da sintaxe do Arduino vieram de conceitos apresentados previamente pelo Projeto de processamento (*Processing Project*) [44], *software* livre (*open-source*) desenvolvido por Casey Reas e Ben Fry com o intuito de ensinar programação com contexto visual. A ideia era facilitar o desenvolvimento de projetos eletrônicos para aqueles que possuem pouca experiência com eletrônica e programação, como artistas e designers [46]. Considerando as semelhanças nos objetivos dos projetos, faz sentido que tenham utilizado como base da IDE a desenvolvida pelo Projeto de Processamento.

Monitor Serial e Serial Plotter

A IDE, além de oferecer um local para o código ser escrito, oferece ferramentas que facilitam o desenvolvimento do código e a análise da troca de informações entre o *hardware* e o *software*. Entre as que se destacam, está o monitor serial (*Serial Monitor*), que é utilizado para observar a comunicação do computador com a placa Arduino, fornecendo informações em tempo real dos dados que estão sendo trocados e sendo bastante útil para entender o que está acontecendo durante a execução de um programa[45]. Existe também o Serial Plotter que gera gráficos em tempo real dos dados seriais (*serial data*), facilitando a análise dos dados [45].

Bibliotecas

Outra aplicabilidade interessante da IDE é o fácil acesso às bibliotecas (*Libraries*), blocos de código que oferecem ações adicionais para o programador. São facilitadores na hora de montar um programa, já que oferecem funcionalidades prontas que, de outra forma, teriam que ser implementadas manualmente. Exemplos seriam bibliotecas que permitem controlar um motor ou para fazer a comunicação com um módulo *bluetooth* com apenas algumas funções. Elas podem ser oficiais, feitas pela empresa do Arduino, ou desenvolvidas por outras pessoas, já que o software é livre, gerando assim uma grande quantidade de bibliotecas disponíveis. Essas podem ser importadas pelo menu da IDE, onde existe um banco com as bibliotecas mais famosas, ou adicionando os arquivos direto na pasta de bibliotecas do Arduino [47]. A página de gerenciamento de bibliotecas pode ser visto na Figura 3.2

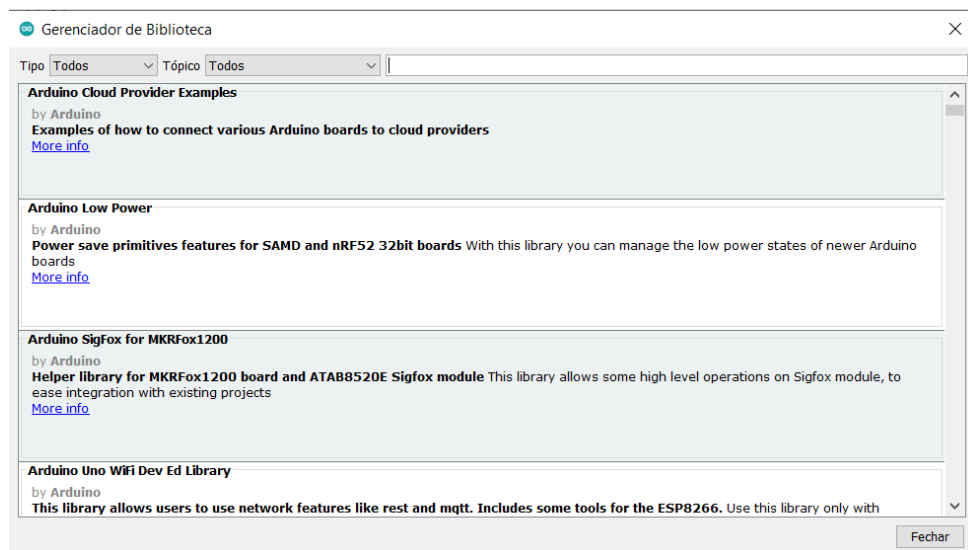


Figura 3.2: Gerenciador de bibliotecas do Arduino.

3.1.2 Linguagem de Programação do Arduino

É a sintaxe utilizada para programar e manipular a placa e seus Atributos dentro da IDE, é a forma como as instruções devem ser escritas para serem corretamente interpretadas e posteriormente executadas pelo Arduino. Um programa escrito na Arduino IDE, que utiliza a linguagem de programação do Arduino, se chama *Sketch* [48].

A linguagem de programação do Arduino utilizou como base o projeto de tese de Hernando Barragan sobre a placa de fiação (*Wiring Board*), cujo objetivo era oferecer uma forma de artistas e designers trabalharem com eletrônicos de maneira mais abstrata e menos complicada, oferecendo uma linguagem simples, mais próxima da humana, que pudesse controlar os componentes eletrônicos sem se aprofundar neles, em poucas linhas de código [49].

A linguagem do Arduino é baseada em C++ [50], linguagem de programação orientada a objeto bastante popular, adicionando novos métodos e funções que fazem sentido para o contexto do projeto. De maneira geral, esses métodos servem para trocar informações com a placa, definindo um pino como entrada ou saída, digital ou analógica e lendo ou escrevendo valores para esses pinos, permitindo a fácil integração de componentes eletrônicos como LEDs, motores e sensores, fazendo com que em apenas algumas linhas de códigos seja possível enviar comandos para o LED acender e o motor girar ou receber dados possivelmente capturados pelo sensor de movimento.

Um código em linguagem de Arduino possui uma estrutura base que consiste no cabeçalho, com as importações de bibliotecas e outros arquivos, a declaração das variáveis que serão utilizadas e duas funções, uma chamada **setup()**, que é executada apenas no início do programa, e outra **loop()**, que será executada depois do **setup()** e continuará sendo executada até que algum outro comando diga o contrário. Essa estrutura pode ser verificada na Figura 3.3

Alguns exemplos mais importantes de funções da linguagem de programação do Arduino são as funções digitais **digitalRead()**, **digitalWrite()**, **pinMode()**, e as funções analógicas **analogRead()** e **analogWrite()**. [51]

- **pinMode()**: Configura um pino específico da placa como entrada (*INPUT*) ou saída (*OUTPUT*). **pinMode(8, OUTPUT)**, configura o pino 8 para ser uma saída. [52]
- **digitalRead()**: Lê o valor em uma entrada específica da placa, esse valor pode ser *HIGH*(1) ou *LOW*(0). **digitalRead(5)**, lê o pino 5, que deve ser anteriormente configurado como entrada, e retorna seu valor. [53]
- **digitalWrite()**: Escreve o valor (*HIGH* ou *LOW*) em uma saída específica da placa. **digitalWrite(3, HIGH)**, escreve no pino 3, que deve ser previamente configurado como saída, o valor *HIGH* (1). [54]

```

#include <SoftwareSerial.h>

int led = 4;

void setup() {
  pinMode(led, OUTPUT);
}

void loop() {
  digitalWrite(led, HIGH);
}

```

Figura 3.3: Linguagem de programação Arduino.

- `analogRead()`: Lê um valor analógico de uma entrada específica da placa (PWM). Esse valor passa por um conversor de analógico para digital de 10bits que traduz voltagens de 0 até a voltagem de operação (5V ou 3.3V) em valores analógicos (“0” a “1023”). `analogRead(3)` lê o pino 3 e retorna seu valor convertido para um valor digital. [55]
- `analogWrite()`: Escreve um valor analógico em uma saída específica da placa (PWM). `analogWrite(3, 50)`, escreve no pino 3, que deve ser previamente configurado como saída, o valor 50. [56]

3.2 O Hardware

As placas Arduino (*Arduino Boards*) são os componentes físicos que interagem com a IDE e que realizam os comandos dados pelos *Sketches*, são a parte de *hardware* do Arduino. De maneira geral, elas possuem um microcontrolador que recebe e executa as instruções a serem seguidas, e diversos pinos que servirão para fazer a interação da placa com outros componentes eletrônicos. Os pinos podem ler entradas (*inputs*): sensores de movimento, de distância ou botões e também enviar dados como saídas (*outputs*): ligar um LED ou um motor [44]. As placas se comunicam com o computador pela entrada USB.

Existem diversos modelos de placas Arduino, cada uma com suas individualidades e projetadas pensando em necessidades específicas para diferentes projetos. O modelo tradicional se chama Uno (Arduino Uno) e é o mais comum para projetos, já que oferece uma boa quantidade de pinos, um bom tamanho e boa performance. Existem também modelos como o Arduino Nano que possui praticamente a mesma especificação, mas em uma placa mais compacta [57], e o Arduino Mega, que é maior, possuindo mais pinos e mais funcionalidades [3].

As placas de Arduino possuem em geral os mesmos componentes, um microcontrolador, pinos digitais de entrada/saída (dos quais alguns podem ser utilizados como saídas PWM e outros como comunicação Serial), pinos de entrada analógica, um cristal oscilador (16 Mhz *Crystal oscillator*), uma entrada USB, que pode ser utilizada como porta serial ou de energia, uma entrada P4, um cabeçalho ICSP (*ICSP header*) e um botão de resetar (*reset*). Algumas placas possuem componentes a mais ou a menos, buscando se adaptar a diferentes necessidades. [4]

3.2.1 Microcontrolador

Um microcontrolador é um circuito integrado composto de um processador, memória e periféricos de entrada e saída. Ele funciona como um pequeno computador, sendo otimizado para realizar certas atividades predefinidas e controlar e se comunicar facilmente com dispositivos analógicos e digitais. [58]

É o microcontrolador que recebe as instruções a serem executadas e que se comunica com os periféricos, recebendo e enviando dados.

Diferentes placas Arduino possuem diferentes microcontroladores. O mais comum, que é o utilizado no Arduino Uno, é o ATmega328P. [4]

3.2.2 Entradas e Saídas Digitais

Os pinos digitais têm esse nome por atuarem com grandezas digitais, ou seja, de valores definidos, que não variam de forma contínua com o tempo. Sinais digitais possuem possibilidades finitas de valores, no caso dos pinos digitais do Arduino os valores possíveis são 0 (*LOW*) e 1 (*HIGH*), que representam a ausência ou presença de corrente, respectivamente. Os pinos operam com a tensão de operação de 5V e são definidos por padrão como entrada (*INPUT*), e caso seja necessário utilizá-los como saída, deve ser definido utilizando a função `pinMode()` passando como atributo o pino e o valor “*OUTPUT*”.

Quando os pinos estão definidos como entradas, se encontram em um estado de alta impedância, ou seja, alta tensão e baixa corrente, resultando na necessidade de uma corrente bem baixa para serem ativados, fazendo com que diversas funções como monitorar

botões e alguns sensores seja extremamente fácil. Um fator a se levar em conta é que pinos definidos como saída digital, caso não estejam conectados a algo, podem sofrer variações em seu estado por ruído elétrico no ambiente. Para resolver essa questão é possível adicionar um resistor conectando a entrada ao terra (*pulldown resistor*) ou a +5V (*pullup resistor*), que vão colocar a entrada em 0 (*LOW*) ou 1 (*HIGH*) respectivamente quando não houver um sinal definido de entrada. Esse efeito é atingido utilizando uma resistência bem alta, que fará com que, no caso de alguma outra tensão existir, a conectada ao “+5V” ou ao terra pelo resistor seja ignorada [59].

Os pinos definidos como saídas ficam em estado de baixa impedância, ou seja, possuem baixa voltagem e alta corrente, permitindo que esses pinos tenham boa capacidade de alimentar outros circuitos, podendo oferecer correntes de -40mA até +40mA. Essa corrente é suficiente para facilmente alimentar certos componentes eletrônicos, como LEDs, *buzzers*, e alguns sensores, mas não é capaz de alimentar componentes que necessitam de maior amperagem, como motores em geral, sendo necessário alguma fonte de alimentação externa. [59]

Alguns desses pinos cumprem a função de realizar a comunicação serial com o computador e outros dispositivos, sendo chamados de portas Serial (UART). O Arduino Uno possui apenas uma porta serial, que utiliza os pinos “0” e “1” para receber (RX) e mandar (TX) dados seriais respectivamente, sendo eles os responsáveis por fazer a comunicação com o computador. Não é recomendado conectar algo nesses pinos, já que pode interferir na comunicação. Nesse caso, seria interessante utilizar um Arduino Mega, que possui mais uma porta serial, que utiliza os pinos “19” e “18” para receber e mandar dados respectivamente. [60]

3.2.3 Saídas PWM

As saídas PWM (*Pulse Width Modulation*) são pinos da placa Arduino que possuem a capacidade de gerar sinais analógicos a partir de valores digitais, ou seja, valores inteiros de “0” a “255” são transformados em sinais analógicos de “0” a tensão de operação da placa (5V ou 3.3V dependendo do modelo do Arduino). Para fazer essa conversão utiliza-se a técnica PWM, que gera uma onda quadrada com o sinal digital e vai ligando e desligando esse sinal, alterando a quantidade de tempo que ele passa em cada estado, dependendo de qual o valor analógico esperado. O tempo que o sinal passa ligado é chamado de largura de pulso e é quem deve ser alterado para alcançar os diferentes valores analógicos. [2]

A função utilizada para operar esses pinos é a **analogWrite()**, que recebe como argumento o pino PWM e o valor digital, esse valor corresponderá a uma largura de pulso, sendo que “255” seria uma onda reta sempre no valor mais alto e “127” seria uma

onda quadrada que passa metade do ciclo ligada e metade desligada, e assim por diante. As formas das ondas podem ser visualizadas na Figura 3.4.

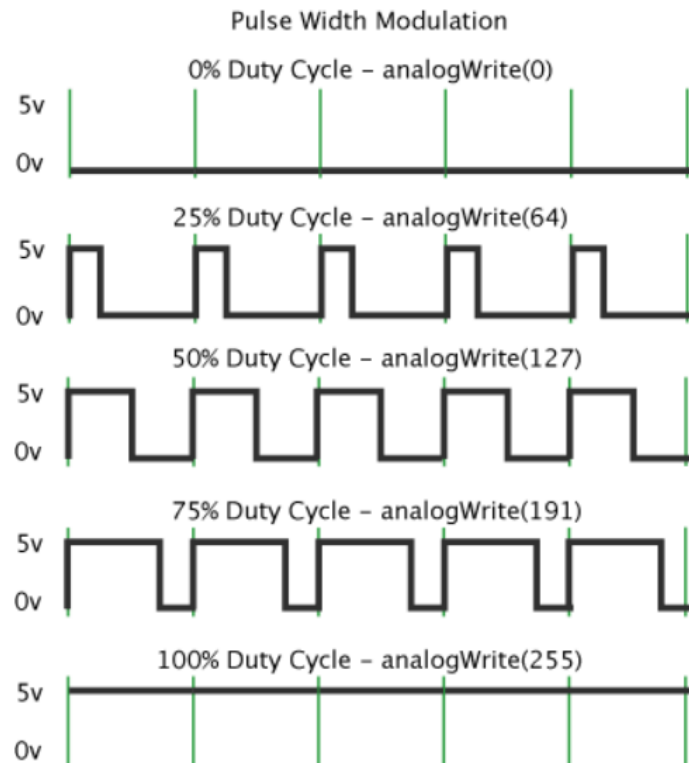


Figura 3.4: Função PWM[2].

3.2.4 Entradas Analógicas

Os pinos de entrada analógica podem servir para os mais diversos usos, desde receber dados de um sensor que indica o quão próximo ele está de algum objeto até a temperatura constatada por um sensor de temperatura. Esses pinos funcionam recebendo tensões entre “0” e a tensão de operação da placa (5V ou 3.3V dependendo do modelo do Arduino) e convertendo esse valor para um valor inteiro de “0” a “1023”. Essa conversão é realizada por um conversor multi-canal de analógico para digital de 10 bits. [61]

A função utilizada normalmente com esses pinos é **`analogRead()`**, que recebe como argumento um pino de saída analógica e já realiza a conversão do valor analógico recebido e retorna o valor digital correspondente. Esses pinos podem funcionar também como entrada ou saída digital sem nenhuma perda, podendo ser tratados como um pino digital no código, sem instruções adicionais. [61]

3.3 Por que o Arduino?

Na confecção desse projeto, diversos fatores foram levados em conta na hora de escolher os componentes eletrônicos que seriam utilizados, principalmente para selecionar a placa com microcontrolador, o Arduino. Dentre eles, os que mais se destacam são:

- Facilidade de reprodução do *Hardware*.

Tendo em vista a importância para esse projeto de ser reproduzido e utilizado pela maior quantidade de pessoas interessadas possível, o Arduino, que é a placa de microcontrolador mais famosa do mundo, se mostrou uma das escolhas mais atraentes, já que pode ser facilmente encontrado em lojas de materiais eletrônicos ou na internet.

- Baixo custo.

O Arduino possui, relativamente, um baixo custo, considerando outras placas presentes no mercado nacional. Apesar de ser um produto importado, a empresa Arduino já existe a vários anos e oferece uma grande variedade de opções de placas. Por possuir uma grande porcentagem do mercado, as placas são produzidas por um preço mais baixo, as tornando bastante competitivas.

- Facilidade de programação

A linguagem de programação do Arduino foi desenvolvida com o objetivo de ser facilmente entendida e que simples comandos pudessem controlar a maioria dos componentes eletrônicos conectados a ela. Esse tipo de linguagem facilita bastante na hora de controlar uma grande variedade de periféricos, e a comunicação com o celular.

- *Open Source*

O Arduino é uma das plataformas *open-source* mais famosas do mundo, possuindo uma quantidade expressiva de usuários, com todos os tipos de proficiência, e em todos os lugares do mundo. Essa grande base de usuários gera uma comunidade que está sempre disposta a participar de novos projetos ou ajudar na confecção de existentes, fazendo com que esse trabalho possa ser futuramente melhorado ou alterado para satisfazer melhor as necessidades de cada indivíduo.

Neste capítulo foram apresentados os conceitos e características importantes da plataforma Arduino. No próximo capítulo serão apresentados os componentes eletrônicos principais utilizados na confecção desse projeto.

Capítulo 4

Componentes Eletrônicos

Para o desenvolvimento do *hardware* foram utilizados diversos componentes eletrônicos, sendo eles:

- Arduinos;
- Ponte H e Motor de Corrente Contínua;
- Motor Servo;
- Módulo *Bluetooth*;
- LEDs;
- Alto Falante;
- *Display* TFT;

As características fundamentais desses componentes serão explicitadas nesse capítulo.

4.1 Arduinos

Durante o desenvolvimento desse projeto surgiu a necessidade de utilizar duas placas Arduino devido a quantidade de pinos que os periféricos eletrônicos ocupam.

Um Arduino Mega fará o papel de controlador principal, estando conectado ao módulo *Bluetooth*, recebendo assim os comandos vindos do aplicativo. O Mega estará conectado também a um Arduino Uno rev3, que receberá instruções do Mega por meio de pinos digitais.

Tanto o Arduino Mega quanto o Uno podem ser alimentados conectados a um computador pela entrada USB (5V) ou por uma alimentação externa utilizando um conector P4, sendo que essa possui um regulador de tensão. No caso de uma alimentação externa,

a tensão de alimentação ideal desses Arduinos é de 7V até 12V, podendo operar com 6V até 20V. Tensões menores do que 7V podem fazer com que a tensão de funcionamento da placa fique instável, não conseguindo se manter em 5V, e maiores que 12V podem danificar o regulador de tensão da placa, já que podem sobreaquecê-lo.

As placas possuem pinos terra (GND), pinos 5V, pino regulador de tensão 3.3V capaz de até 50mA de corrente e pinos VIN, que funcionam como um pino 5V caso a placa esteja alimentada pela porta USB ou então como um pino que terá a mesma tensão que entra pelo P4, da fonte externa, caso ela esteja sendo utilizada ao invés da USB.

4.1.1 Arduino Mega

O Arduino principal desse projeto é o Arduino Mega 2560, ele possui um microcontrolador ATmega2560, 54 pinos digitais de entrada ou saída, dos quais 15 podem ser utilizados como saídas PWM, 16 entradas analógicas, 4 UARTs (*hardware serial ports*), um oscilador de 16Mhz, uma entrada USB, uma entrada de energia, um botão de *reset* e um cabeçalho ICSP (ICSH *header*) [3]. Mais informações sobre esse Arduino podem ser visualizados na Figura 4.1 e na Figura 4.2.

4.1.2 Arduino Uno

O modelo de Arduino tradicional se chama Uno (Arduino Uno) e está na versão Rev3 (Arduino Uno Rev3), ele possui um microcontrolador ATmega328P, 14 pinos digitais, dos quais 6 podem ser usados como saídas PWM, 6 entradas analógicas, um oscilador de 16Mhz, uma entrada USB, uma entrada de energia, um botão de *reset* e um cabeçalho ICSP (ICSP *Header*) [4]. Mais informações sobre esse Arduino podem ser visualizados na Figura 4.3 e na Figura 4.4.

4.2 Ponte H e Motor de Corrente Contínua

A ponte H L298n é um circuito monolítico integrado construído com o intuito de ser capaz de controlar motores de corrente contínua (motor DC), motores de passo, solenoides e outros componentes. Nesse projeto a ponte será usada para controlar dois motores DC.

Como mostrado na Figura 4.5, e seguindo o *Datasheet* do componente [62], a ponte H possui 2 entradas para cada motor, 2 para o MOTOR A e 2 para o MOTOR B, sendo que uma entrada ativa o giro para um lado e a outra para o outro lado. Possui uma entrada de energia (VCC), que aceita até 46V, um terra (GND) e uma porta controladora de tensão 5V que pode ser utilizada para alimentar outro componente.

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

Figura 4.1: Arduino Mega 2560 Datasheet [3].

Os pinos **ENABLEA** e **ENABLEB** são para definir a velocidade de rotação dos motores, eles precisam ser associados a pinos PWM do Arduino e recebem um valor que corresponderá a uma velocidade.

Os pinos **IN1** e **IN2** são para controlar o giro do motor A e o **IN3** e **IN4** o do motor B, sendo que apenas um dos pinos de cada motor pode estar ligado por vez, caso contrário, não ocorre o giro. No caso do motor A, por exemplo, apenas o **IN1** ou o **IN2** deve estar ligado. Alterando qual pino está ligado muda o sentido de rotação do eixo do motor. Esses pinos podem ser ligados em entradas digitais padrões, já que recebem apenas *HIGH*, ligado, ou *LOW*, desligado.

Os motores que serão conectados na ponte H são motores de corrente contínua também conhecidos como motores DC (Figura 4.6). Esses motores são componentes que possuem a capacidade de transformar energia elétrica em energia mecânica.

Ele é composto basicamente por uma bobina, conectada ao eixo do motor sobre a influência de um campo magnético gerado por ímãs. Quando uma corrente passa pela bobina, uma força magnética é gerada, fazendo com que a bobina entre em movimento,

Pinout Diagram

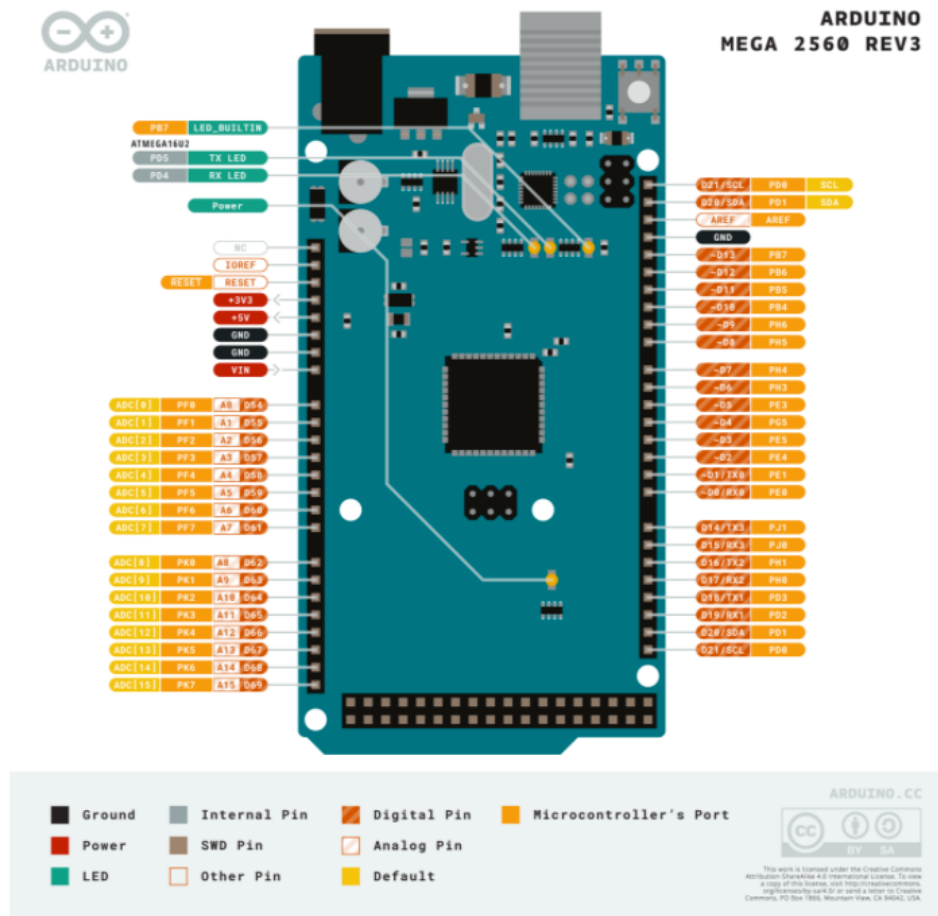


Figura 4.2: Arduino Mega 2560 Pinout [3].

logo, o eixo começa a girar.

4.3 Motor Servo

Um motor servo é um motor que possui a capacidade de controlar, de forma precisa, a posição angular, a velocidade e a aceleração do giro. É basicamente um motor agregado de um componente que detecta a posição do eixo. Esse tipo de motor é extremamente útil para projetos que requerem rotações precisas.

A forma como ele funciona, de maneira simplificada, é utilizando sensores para verificar a posição, torque ou velocidade do eixo. Esses dados são re-alimentados para o motor com o intuito de verificar a diferença entre esse estado atual do motor e o estado que a instrução quer que ele esteja, realizando, em seguida, a movimentação. [63]

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

Figura 4.3: Arduino Uno Rev3 Datasheet [4].

Os motores servo possuem 3 pinos de conexão. Um para o terra, um para o Vcc e um para a conexão com o Arduino(Figura 4.7).

4.4 Módulo *Bluetooth*

O módulo *Bluetooth* HC-05/ZS-040 será o responsável por realizar a conexão entre o Arduino e o celular com o aplicativo (Figura 4.8).

O módulo possui 6 pinos:

- VCC: Alimentação elétrica.
- GND: Terra.
- RXD: Pino que recebe dados.

Pinout Diagram

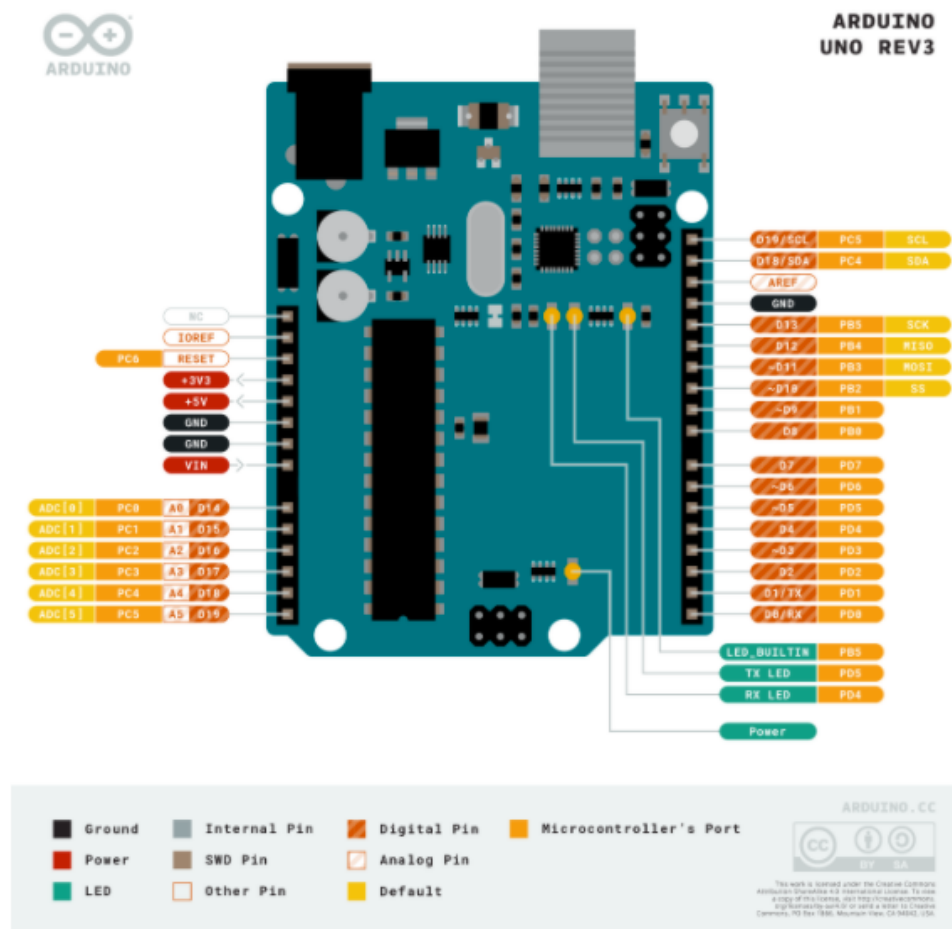


Figura 4.4: Arduino Uno Rev3 Pinout [4].

- TXD: Pino que envia dados.
- EN: Se ligado aciona o modo mestre, que é utilizado para conexão com outros *bluetooths*.
- STATE: Indica se o *bluetooth* está conectado, *HIGH* se conectado, *LOW* caso contrário.

4.5 LEDs

Light emitting diode (LED) ou diodo emissor de luz é um componente eletrônico que pode emitir luz visível, que ocorre quando o diodo é energizado. Ele têm baixo consumo de energia e alta eficiência, além de ser um componente de baixo custo. [64]

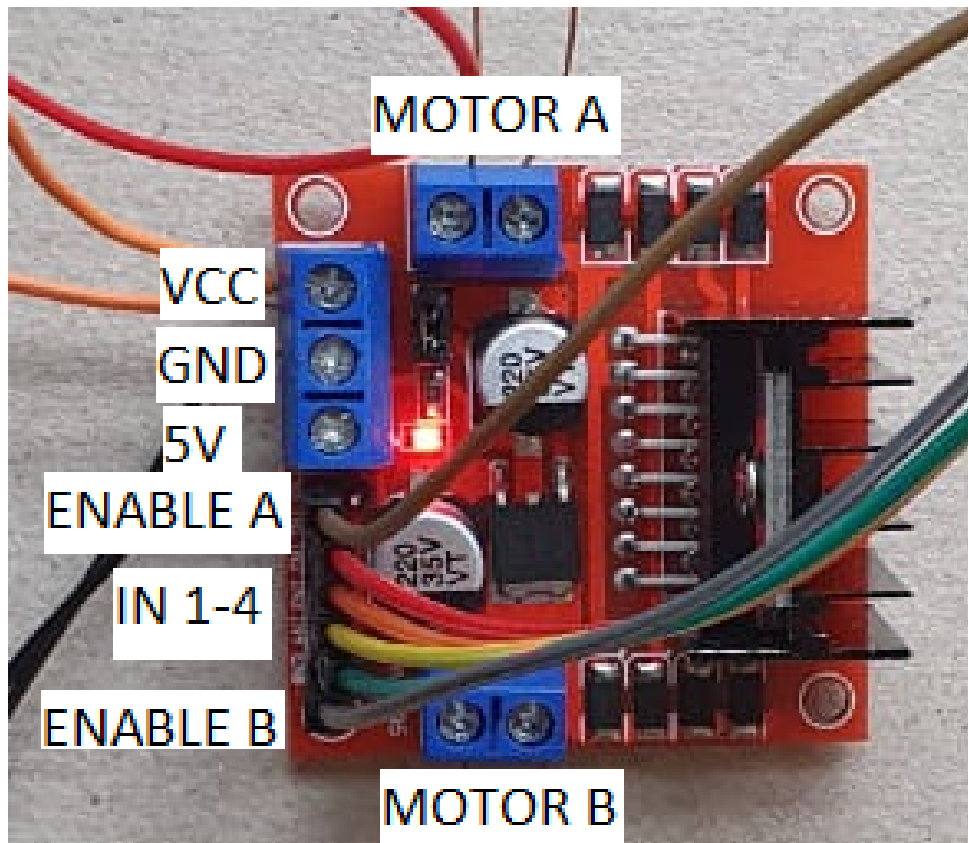


Figura 4.5: Ponte H Pinout.



Figura 4.6: Motor DC.

Eles possuem dois pinos, duas "pernas", uma maior, positiva, que se conectará com o Arduino e uma menor, negativa, que irá para o terra (Figura 4.9).

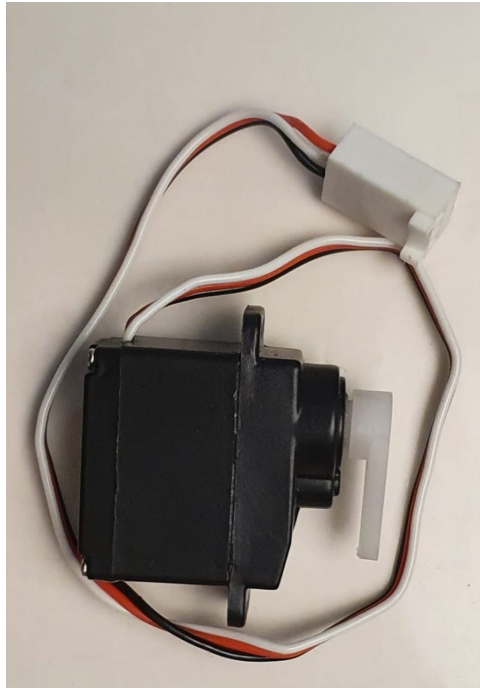


Figura 4.7: Motor Servo .



Figura 4.8: Módulo Bluetooth.

4.6 Alto Falante

O alto falante é um componente capaz de converter energia elétrica em energia mecânica, que irá comprimir o ar e gerar som. Ele possui dois pinos, um que irá receber dados e outro que será conectado com o terra.

O Arduino irá gerar uma onda quadrada de acordo com a melodia desejada a ser

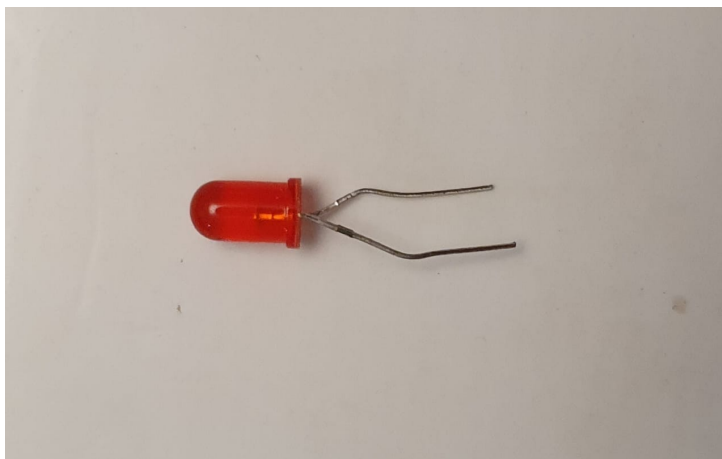


Figura 4.9: LED Vermelho .

tocada, essa onda será enviada para o alto falante, onde a corrente irá passar por uma bobina gerando um campo magnético que irá interagir com o campo magnético de um ímã preso ao cone exterior do alto falante. A interação desses campos magnéticos irá repelir ou atrair o ímã com diferentes intensidades, dependendo de como é a onda que será reproduzida, fazendo o movimento característico do alto falante. O movimento do ímã cria uma pressão no ar gerando o som.[65]

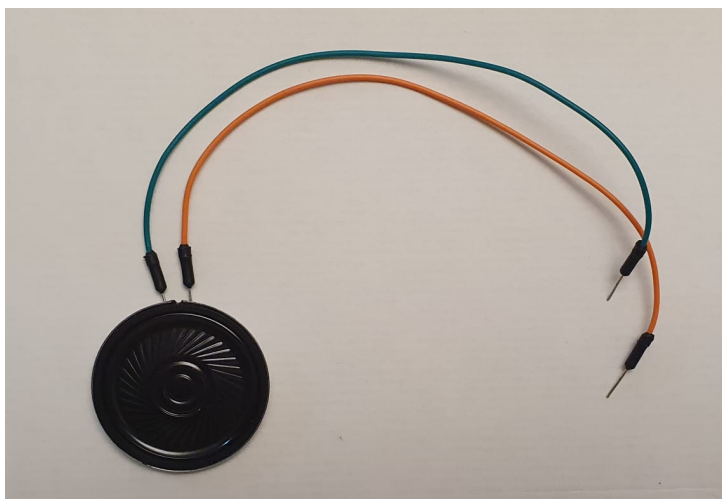


Figura 4.10: Alto Falante .

4.7 *Display* TFT

O *display* TFT ILI9341 (Figura 4.12) é uma tela de cristal líquido sensível ao toque de 240x320 *pixels* e 18 *bits* de cores e é capaz de exibir 262 mil variações de cor. Possui ainda

um módulo cartão micro SD embutido.[66]

O *display* TFT é um *shield* para Arduino, isso significa que ele encaixa em cima da placa, utilizando todos os pinos de um Arduino Uno. Os pinos do *display* podem ser visualizados na Figura 4.11

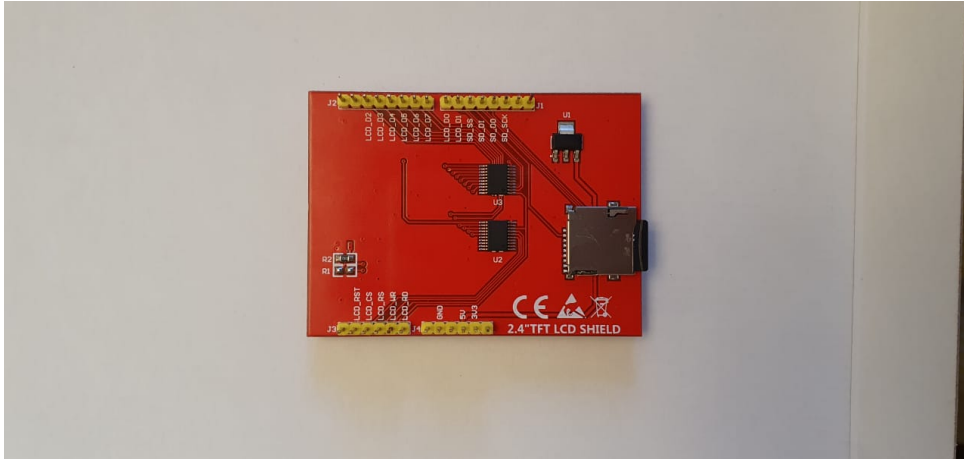


Figura 4.11: *Display* TFT Pinout .

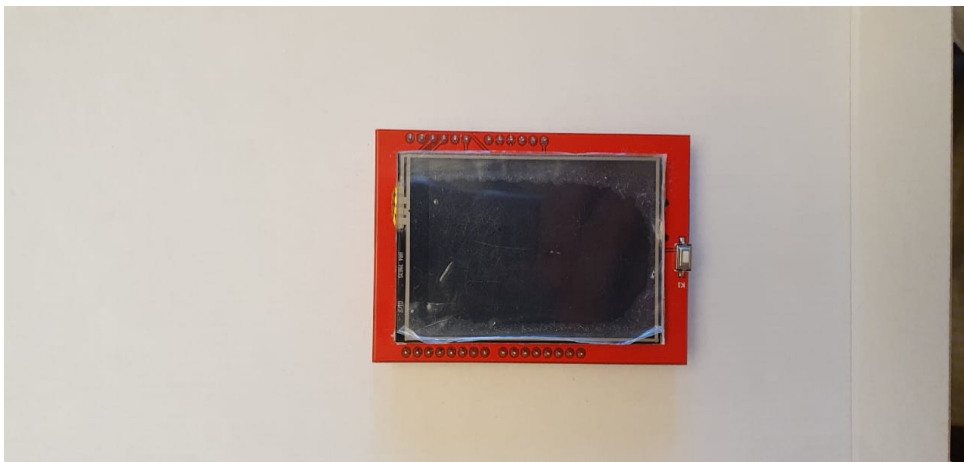


Figura 4.12: *Display* TFT .

Neste capítulo foram apresentados os conceitos e características importantes dos componentes eletrônicos utilizados. No próximo capítulo, será explicado como os componentes foram integrados, como foi feita a montagem do Cubinho e as possibilidades de funções que ele pode desempenhar.

Capítulo 5

Testes dos Componentes

Nesse capítulo serão descritos os testes feitos com os componentes eletrônicos com o intuito de verificar o correto funcionamento desses módulos.

5.1 Arduinos

Os testes de ambos Arduinos usaram o mesmo código, chamado **Fade**, que pode ser encontrado pela IDE do Arduino clicando em Arquivo ->Exemplos ->Basics ->Fade ou no site oficial do Arduino [5]. O esquemático da montagem do circuito está na Figura 5.1.

O código (Figura 5.2) inicia com a função **setup()**, onde o pino 9 é definido como sendo saída, nele será conectada o LED.

A função **loop()** inicia com um **analogWrite()** para o pino do LED do valor presente na variável **brightness**, que inicia como 0 e é incrementada a cada giro do *loop*. Este incremento é realizado somando o valor presente na variável **fadeAmount**, que inicia como “5” e é invertido toda vez que **brightness** atinge “255”, virando “-5”, ou quando chega a “0”, voltando a ser “5”.

Em ambos os Arduinos os testes foram bem sucedidos, atingindo o resultado esperado: O brilho do LED vai aumentando até chegar no máximo, quando começará a diminuir até apagar, e iniciará esse processo novamente.

5.2 Ponte H e Motor de Corrente Continua

Na Figura 5.3 está o código de teste utilizado para a ponte H e os motores DC. O esquemático do circuito pode ser visto na Figura 5.4.

No teste, os pinos 6 e 7 serão utilizados para controlar o motor A e os 8 e 9 para controlar o motor B. Esses pinos são definidos como saídas na função **setup()**.

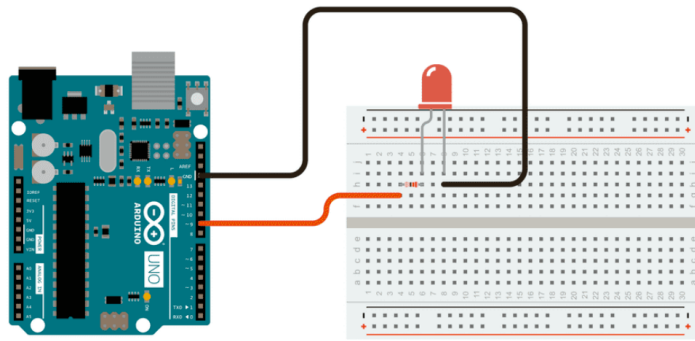


Figura 5.1: Esquemático do circuito de teste dos Arduinos e LED [5] .

```

int led = 9;           // the PWM pin the LED is attached to
int brightness = 0;   // how bright the LED is
int fadeAmount = 5;   // how many points to fade the LED by

// the setup routine runs once when you press reset:
void setup() {
  // declare pin 9 to be an output:
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  // set the brightness of pin 9:
  analogWrite(led, brightness);

  // change the brightness for next time through the loop:
  brightness = brightness + fadeAmount;

  // reverse the direction of the fading at the ends of the fade:
  if (brightness <= 0 || brightness >= 255) {
    fadeAmount = -fadeAmount;
  }
  // wait for 30 milliseconds to see the dimming effect
  delay(30);
}

```

Figura 5.2: Código de teste para os Arduinos [5] .

Dentro da função `loop()` é feito o movimento de rotação dos motores, sendo que apenas um dos pinos de cada motor é ativado, já que o objetivo é que eles se movimentem.

Os testes ocorreram conforme o esperado, os motores giraram de maneira correta.

5.3 Motor Servo

Esse projeto requer o uso de dois motores servo. Para o teste desses componentes foi utilizado o código de exemplo `Sweep` da biblioteca oficial para motores servo do Arduino chamada `Servo.h`. Esse exemplo pode ser encontrado pela IDE do Arduino clicando em


```

int in1 = 6;
int in2 = 7;
int in3 = 8;
int in4 = 9;

void setup() {
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in3, OUTPUT);
}

void loop() {
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  digitalWrite(in3, HIGH);
  digitalWrite(in4, LOW);
  delay(1000);
}

```

Figura 5.3: Código de teste para a ponte H e os motores DC.

Arquivo->Exemplos->Servo->Sweep ou pelo site oficial do Arduino [6]. O esquemático da montagem do circuito está na Figura 5.5.

O código (Figura 5.6) inicia com a inclusão da biblioteca **Servo.h** seguido da criação de um objeto **Servo**, que irá receber como parâmetro o pino onde o motor servo estará conectado, e fará as configurações necessárias para facilitar o controle do motor. Na sequência vêm a declaração da variável **pos**, que representa a posição do eixo do motor. Dentro da função de **setup()** o pino 9 é configurado para a instância do objeto **Servo**, chamado **myservo**. Dentro da função **loop()** temos um laço que comanda o servo a fazer um giro de 180° para um lado e em seguida, no laço seguinte, volta a posição inicial, encerrando o **loop**. Em seguida ele recomeça esse processo e vai fazendo a movimentação de um lado para o outro.

Ambos os motores foram testados e atingiram os resultados esperados.

5.4 Módulo Bluetooth

O código de teste (Figura 5.8) para o módulo *bluetooth* consiste em importar a biblioteca oficial do Arduino **SoftwareSerial.h**, definir o *bluetooth* passando como parâmetros o

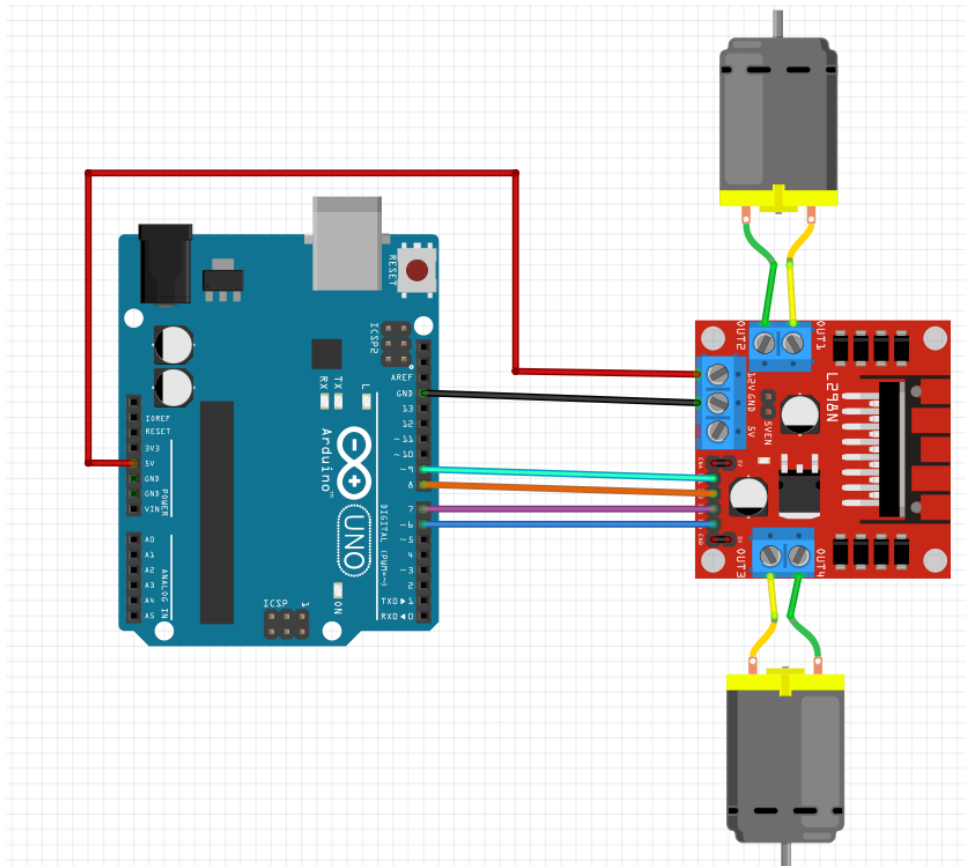


Figura 5.4: Esquemático do circuito de teste para a ponte h e os motores DC.

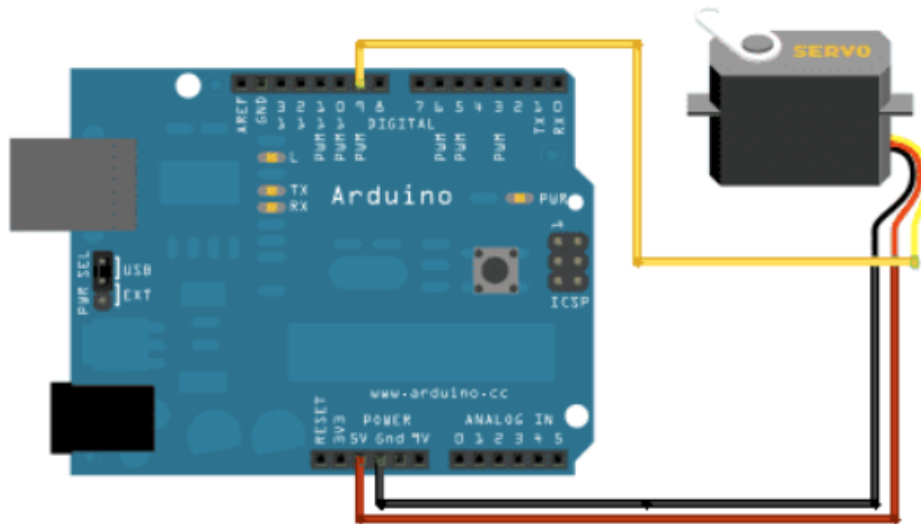


Figura 5.5: Esquemático do circuito de testes do motor servo [6].

pino que recebe os dados do *bluetooth* e o que envia dados ao *bluetooth* e então verificar, a cada volta do laço, se existe algum valor para ser lido do *bluetooth*, caso exista, compara

```

#include <Servo.h>

Servo myservo; // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position

void setup() {
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
}

```

Figura 5.6: Código de teste do motor servo .

esse valor com “1”, sendo verdadeiro, acende o LED.

O esquemático do circuito de teste está na Figura 5.7. Um divisor de tensão foi utilizado no pino de entrada de dados do *bluetooth*, esse divisor serve para diminuir a tensão que chegará ao pino do módulo, já que o Arduino utiliza 5V, uma tensão maior do que a ideal para a alimentação do HC-05, que é 3.3V. Esse divisor de tensão é composto de um resistor de 2.2k Ω ligado ao terra e um de 1k Ω ligado ao pino do Arduino, isso faz com que a tensão que chega ao módulo seja reduzida para aproximadamente o valor ideal.

O teste ocorreu utilizando um aplicativo de comunicação serial para Android e resultou no acendimento do LED como esperado.

5.5 LEDs

Na Figura 5.9 está o código de teste para os LEDs, ele consiste em acender um LED conectado ao pino 9 do Arduino durante 1 segundo, o desligando em seguida e esperando mais 1 segundo, reiniciando assim o laço.

Na Figura 5.1 está ilustrado o esquemático do circuito. O resultado dos testes foi o esperado, o led acendeu e apagou no tempo definido.

5.6 Alto Falante

O código do teste do alto falante está na Figura 5.10 e consiste em tocar uma nota no alto falante utilizando a função **tone()** do Arduino, que recebe como parâmetro o pino

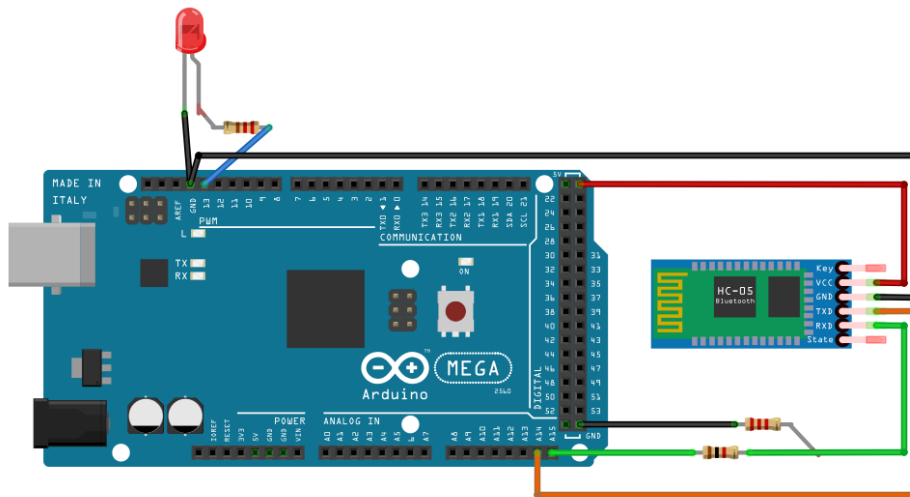


Figura 5.7: Esquemático do circuito de teste para o módulo bluetooth.

```
#include <SoftwareSerial.h>

int led = 13;

int pinReceiveBlu = 68;
int pinSendBlu = 69;

SoftwareSerial bluetooth(pinReceiveBlu, pinSendBlu); // RX, TX do Arduino

void setup() {
  pinMode(led, OUTPUT);
  bluetooth.begin(9600);
}

void loop() {
  if (bluetooth.available() > 0) {
    if (bluetooth.read() == '1') {
      digitalWrite(led, HIGH);
    }
  }
}
```

Figura 5.8: Código de teste para o módulo bluetooth.

de onde está o alto falante e a nota a ser tocada.

O Esquemático do circuito está na Figura 5.11 e o teste funcionou como esperado, a nota tocou durante 1 segundo.

```

#include <SoftwareSerial.h>

int led = 9;

void setup() {
  pinMode(led, OUTPUT);
}

void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}

```

Figura 5.9: Código de teste para os LEDs.

```

int speaker = 13;

int NOTE_G1 = 49;

void setup() {
  tone(speaker, NOTE_G1);
  delay(1000);
  noTone(speaker);
}

void loop() {
}

```

Figura 5.10: Código de teste para o Alto Falante.

5.7 Display TFT

Para realização dos testes do *display* TFT foi utilizado um código de exemplo adaptado da biblioteca **Adafruit_TFTLCD** chamado **tftbmp_shield**, que pode ser encontrado na IDE do Arduino em Arquivo->Exemplos->Adafruit_TFTLCD->tftbmp_shield após importar essa biblioteca ou na página do Github da Adafruit[67]

O código (Figura 5.12) consiste em importar as bibliotecas necessárias para o uso do *display* TFT, definir a *tft*, que é um *shield*, logo não precisa de indicação de quais são

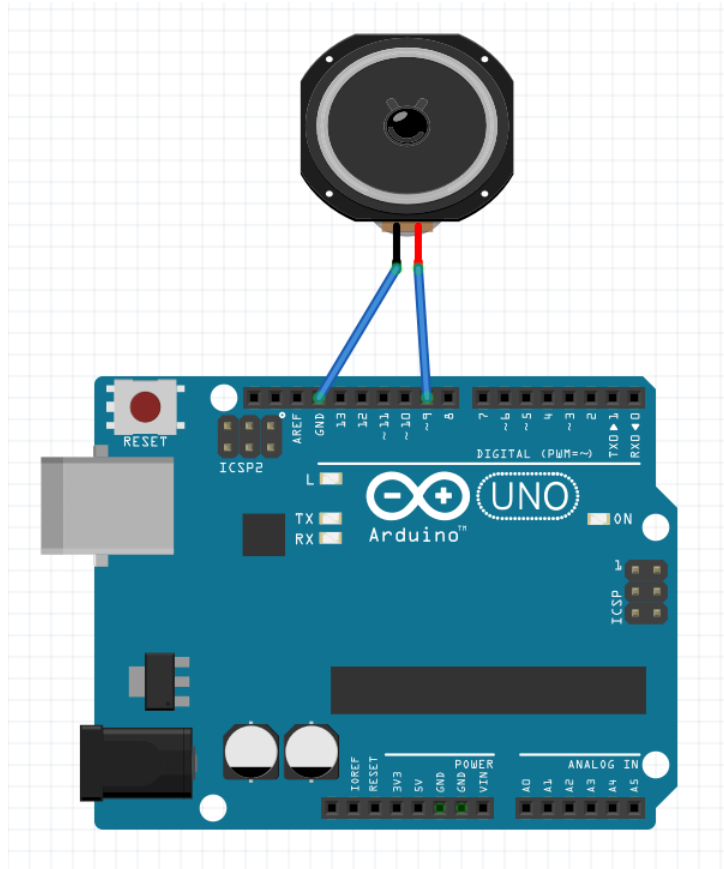


Figura 5.11: Esquemático do circuito de teste para o Alto Falante.

os pinos utilizados, já que encaixará no Arduino e utilizará sempre os mesmos pinos. Em seguida é inicializado o cartão SD e, caso não ocorra nenhum erro, a imagem será desenhada na tela.

O teste ocorreu como esperado, a imagem foi desenhada na tela de maneira correta.

```

#include <Adafruit_GFX.h>
#include <Adafruit_TFTLCD.h>
#include <SD.h>

#define SD_CS 5

Adafruit_TFTLCD tft;
uint8_t spi_save;

void setup()
{
  Serial.begin(9600);

  tft.reset();

  uint16_t identifier = tft.readID();

  tft.begin(0x9341);

  Serial.print(F("Iniciando cartão SD..."));
  if (!SD.begin(SD_CS)) {
    Serial.println(F("falhou!"));
    return;
  }
  Serial.println(F("OK!"));
  spi_save = SPICR;

  bmpDraw("bola.bmp", 0, 0);
  delay(1000);
}

void loop()
{
}

```

Figura 5.12: Código de teste para o o display TFT.

Capítulo 6

O Hardware Desenvolvido

Neste capítulo será relatada a montagem do Cubinho, demonstrando como foi atingido o resultado da Figura 6.18, desde a integração do Arduino como os componentes eletrônicos, sua montagem com a parte externa e a forma como o Cubinho interage com o aplicativo Prevenir.

Os requisitos educacionais foram especificados pela psicopedagoga Maraísa Helena Borges Estevão Pereira, com larga experiência em educação especial: escolha da temática, seleção de funcionalidades e verificação final do produto.

6.1 Montagem dos Componentes

A montagem dos componentes se deu de forma modularizada, iniciando com o circuito que contém cada um dos periféricos e em seguida os adicionando ao Arduino. Um esquemático em blocos pode ser visualizado na Figura 6.1.

Os componentes utilizados na montagem do Cubinho são:

- 1 Arduino Mega 2560
- 1 Arduino Uno Rev 3
- 1 *Display* TFT ILI9351
- 1 Alto Falante
- 2 Motores Servo
- 2 Motores DC
- 1 Ponte H l298n
- 8 LEDs

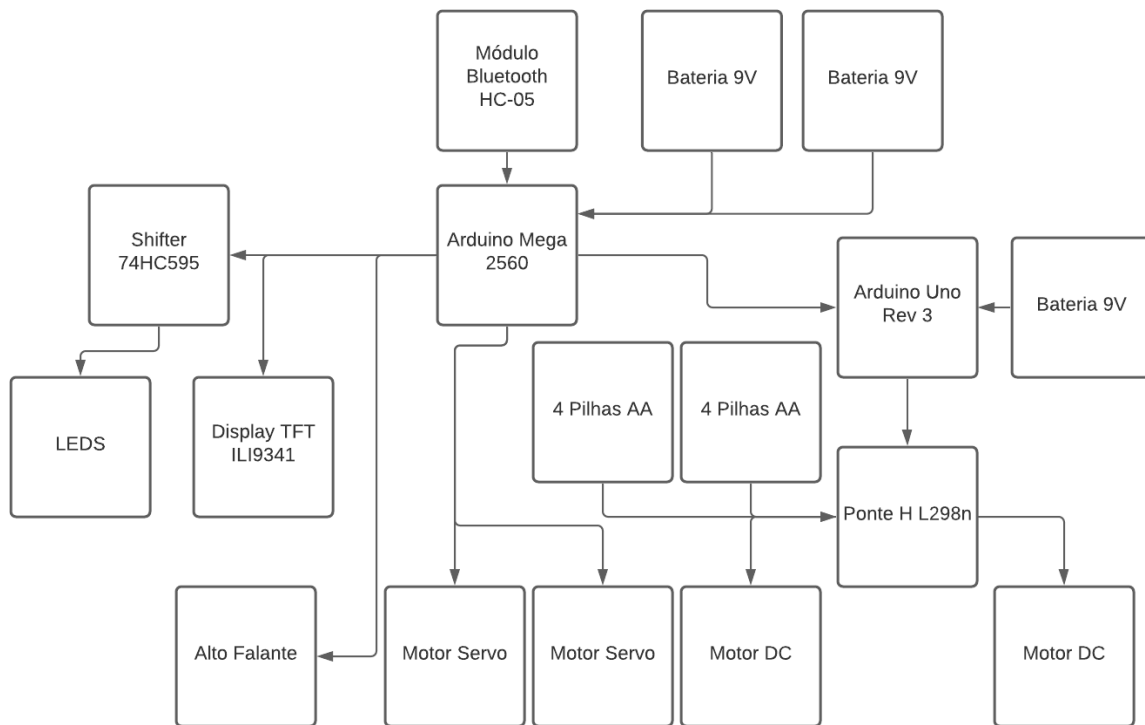


Figura 6.1: Esquemático em blocos do Hardware.

- 1 *Shifter* 74HC595
- 8 Resistores de 220Ω
- 1 Resistor de 1000Ω
- 1 Resistor de 2200Ω
- 1 Módulo *Bluetooth* HC-05
- 3 Baterias 9V
- 8 Pilhas Alcalinas 1.5V
- 2 *Breadboards*

A forma como as ligações foram feitas está ilustrada na Figura 6.2. O alto falante, o *shifter* que controla os LEDs e os 2 motores servo são conectados em entradas digitais do Arduino Mega. O *display* TFT é um *shield*, e portanto encaixa no Arduino Mega sem problemas. O TX e RX do módulo *Bluetooth* utilizam a conexão UART. Pinos digitais também são utilizados para controlar o Arduino Uno, onde são utilizados 4 pinos digitais e 2 PWM para controlar os dois motores DC conectados a ela. Os motores DC devem ser

conectados considerando que os polos do motor devem ficar de maneira oposta na ponte, ou seja, o positivo de um dos motores no primeiro conector do Motor A e o positivo do outro motor no segundo conector do Motor B. O Arduino Mega é energizado por duas baterias de 9V em paralelo, o Arduino Uno, por uma bateria 9V e os motores servo e a ponte H, por 2 conjuntos de 4 pilhas alcalinas AA 1.5V em paralelo.

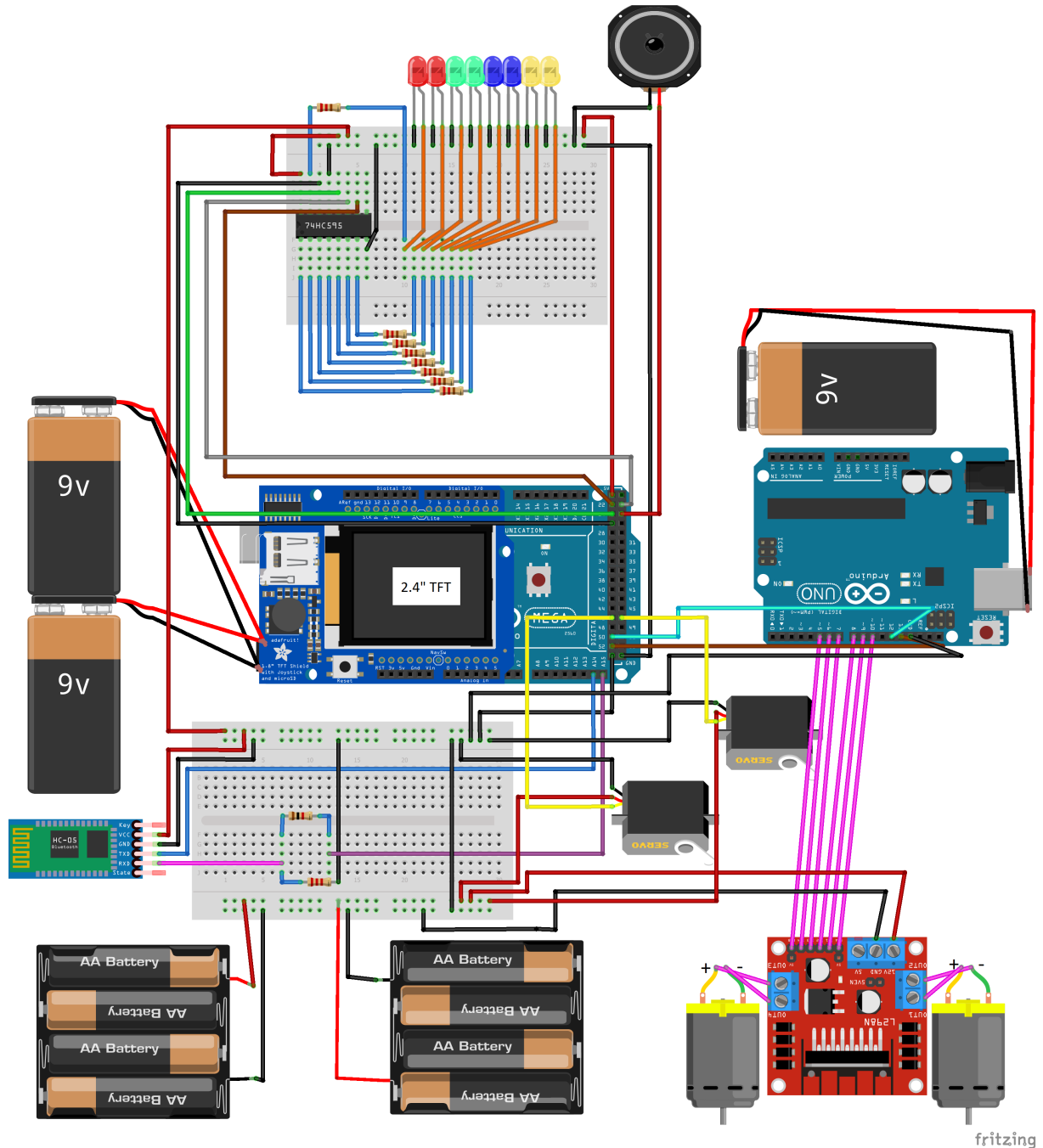


Figura 6.2: Esquemático da montagem dos componentes.

6.1.1 O Cubo

O exterior do Cubinho nada mais é do que um cubo feito de papel panamá em que cada aresta possui 15cm. A escolha do papel panamá se deu devido a facilidade de aquisição desse material e ao seu baixo custo, além da praticidade da montagem. O tamanho das arestas foi definido baseado no tamanho dos componentes que iriam estar em seu interior. Foram cortadas aberturas para encaixar as rodas, os motores servo e o *display* TFT. O cubo pode ser visto desmontado na Figura 6.3.

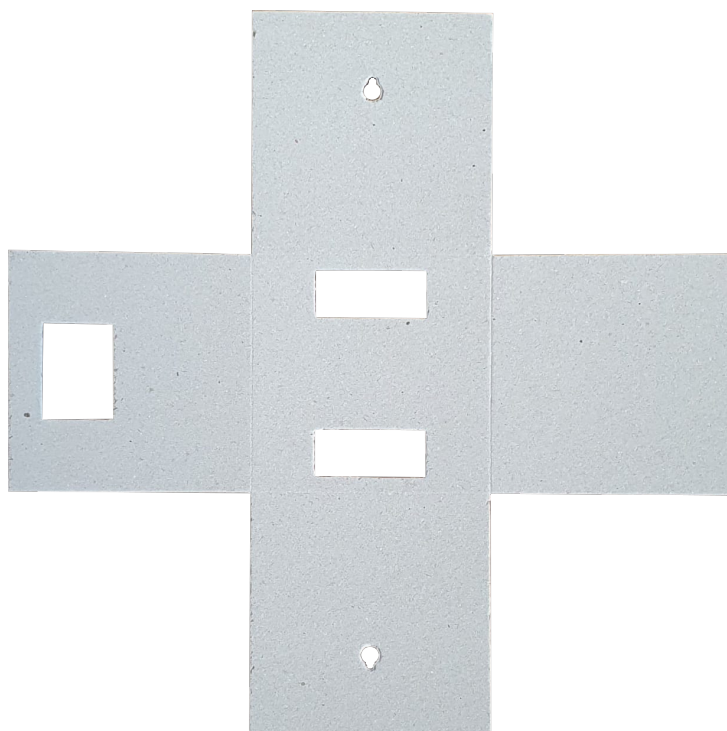


Figura 6.3: Cubo de papel panamá.

6.1.2 A ponte H e os motores DC

O primeiro componente que foi montado foi a ponte H, já que nela estão conectados os motores das rodas, que precisam ficar na parte de baixo do cubo. Para alimentar os motores, foram utilizados dois conjuntos em paralelo de 4 pilhas alcalinas de 1.5V cada, que fornecem um total de 6V ao motor. A ponte H foi conectada ao Arduino Uno, podendo ser visualizada na Figura 6.4.

Os motores foram fixados no cubo utilizando fitas adesivas. As pilhas foram fixadas na parte de trás do cubo e a ponte H foi colocada em cima dos motores, de maneira a ocupar o menor espaço possível e ainda assim não atrapalhar no giro dos motores. Essa parte da montagem pode ser observada na Figura 6.5.

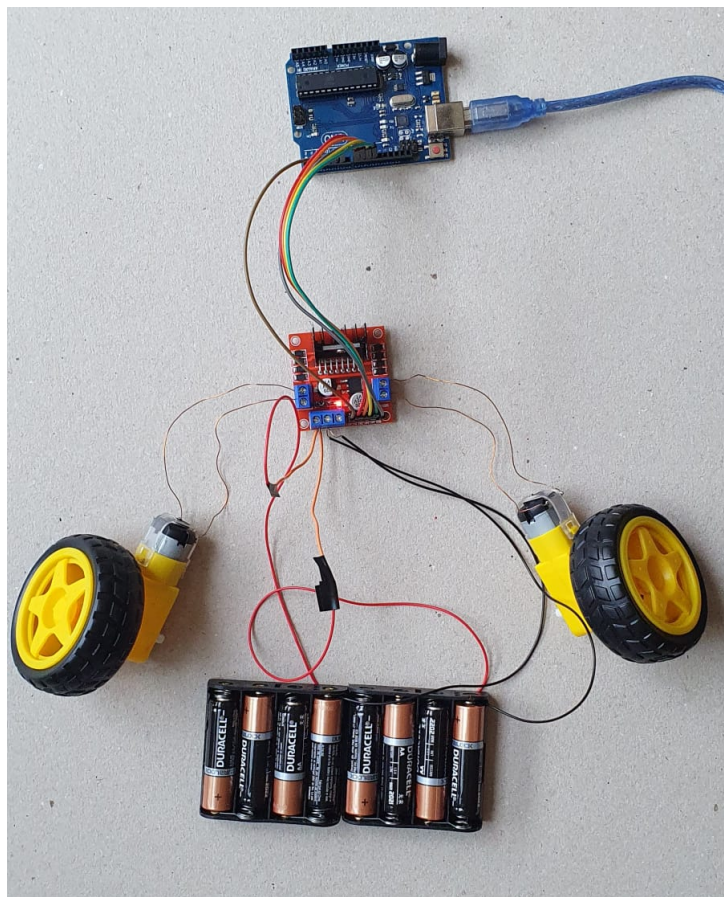


Figura 6.4: Montagem da ponte H e dos motores DC.

6.1.3 Os motores Servo

Dois motores servos foram utilizados para esse projeto e foram inseridos em lados opostos do cubo. Os motores servo serão conectados ao Arduino Mega, que será posteriormente adicionado ao circuito.

Esses motores servo utilizaram as pilhas dos motores DC como fonte de energia, sua montagem pode ser visualizada na Figura 6.6.

6.1.4 O Módulo *Bluetooth*

O módulo *bluetooth* foi inserido em uma *breadboard* (Figura 6.7) para diminuir a dificuldade da montagem e permitir que o divisor de tensão fosse facilmente utilizado.

O módulo foi montado no cubo utilizando uma fita dupla face para grudar a *breadboard* em uma lateral interna do cubo (Figura 6.8).

O módulo *bluetooth* é conectado ao Arduino Mega, que será posteriormente adicionado ao circuito.

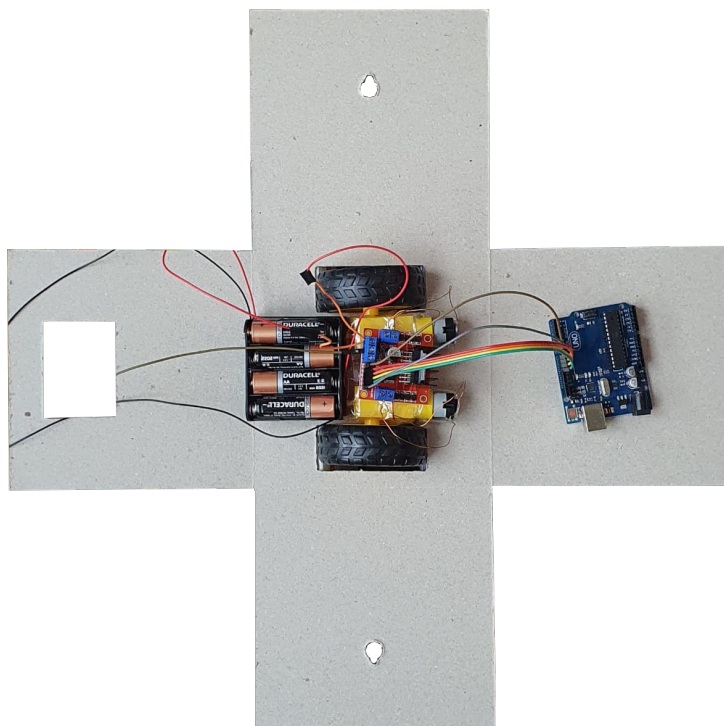


Figura 6.5: Montagem da ponte H e dos motores DC no cubo.

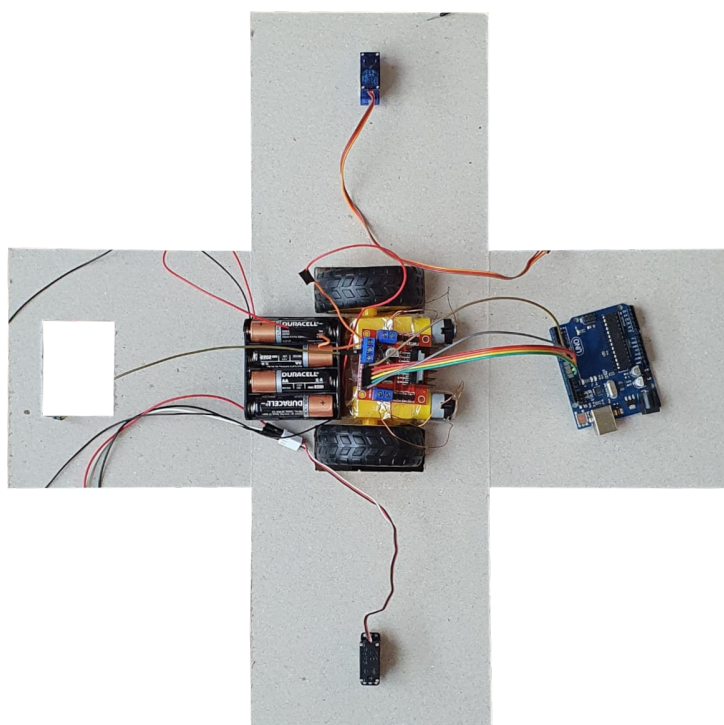


Figura 6.6: Montagem do Motor Servo.

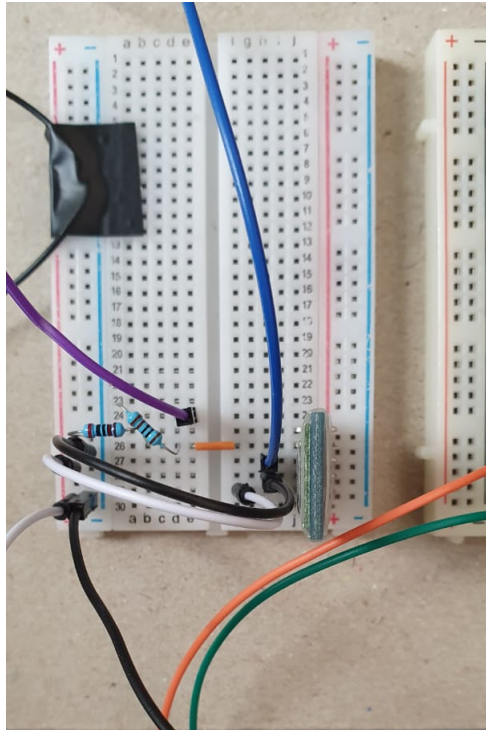


Figura 6.7: Montagem do Módulo Bluetooth.

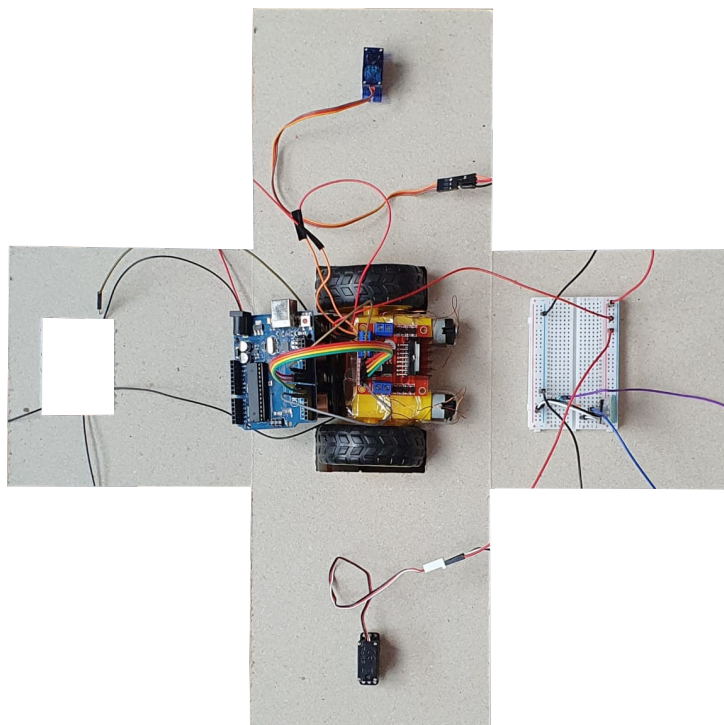


Figura 6.8: Montagem do Módulo Bluetooth no Cubo.

6.1.5 Os LEDs e o Alto Falante

Para esse trabalho, foi utilizado um total de 8 LEDs. Para diminuir a quantidade de pinos utilizados, já que cada LED precisaria de um pino, um *shifter* sn74hc595n (Figura 6.9) foi adicionado.

O *shifter* sn74hc595n consegue transformar 3 entradas seriais em 8 saídas paralelas, uma para cada LED, ou seja, utilizando 3 pinos no Arduino podemos controlar 8 LEDs. A forma como esse *shifter* funciona é deslocando um *bit* em uma sequência de 8 *bits* a cada *clock*, adicionando no início da sequência um novo *bit* que pode ser 0 ou 1. Cada um dos 8 *bits* representa uma entrada, sendo assim, caso queira acionar algo na última entrada, é só adicionar “1” no início dos *bits* e deslocar ele até a posição desejada. [7]

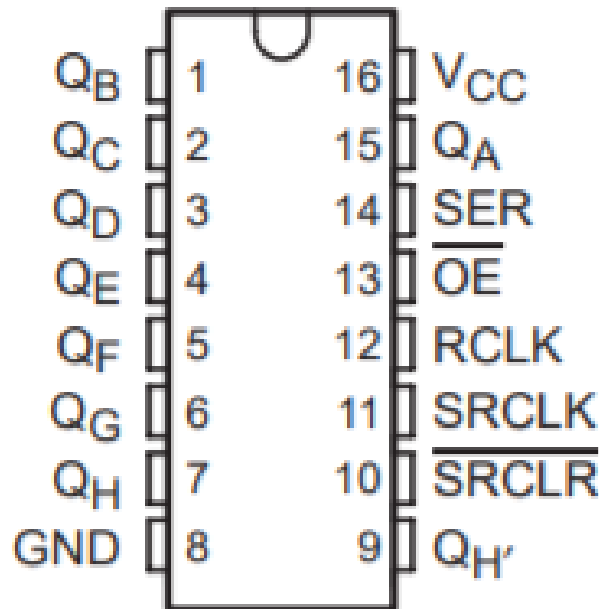


Figura 6.9: sn74hc595n Pinout [7].

O *shifter* foi inserido em uma *breadboard* para facilitar a montagem (Figura 6.10). Os LEDs foram colocados de maneira que a perna negativa fosse conectada direto no terra e a perna positiva a um resistor de 220Ω que então se comunica com alguma saída paralela do *shifter*. Essa resistência é necessária para diminuir a corrente que irá passar pelo LED, que pode ser muito alta e queimá-lo.

O alto falante também foi inserido na *breadboard*, sendo que seu pino negativo foi conectado ao terra e o seu positivo foi conectado ao Arduino.

A *breadboard* foi grudada utilizando fita dupla face em um dos lados internos do cubo. A montagem pode ser verificada na Figura 6.11.

Foi inserido nessa etapa também o Arduino Mega, que recebe, até o momento, as conexões dos motores servo, do módulo *bluetooth*, dos LEDs e do alto falante, além de se comunicar também por *jumpers* com o Arduino Uno.

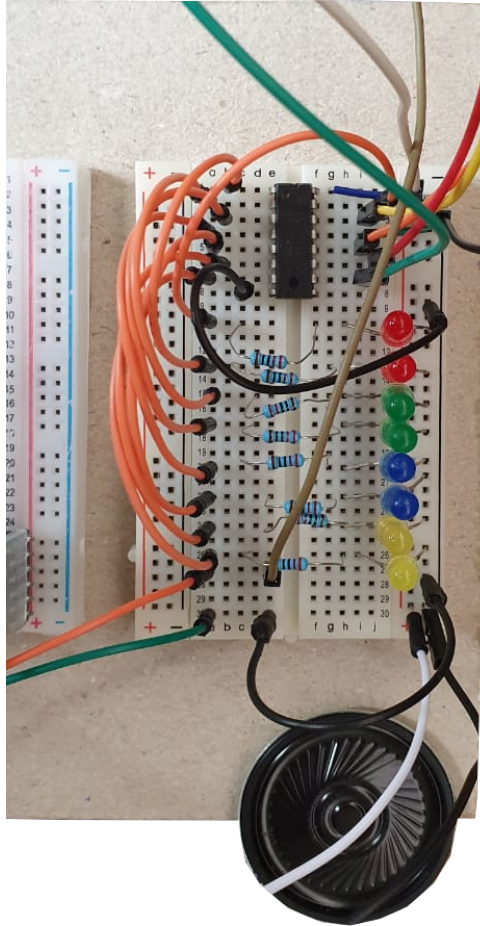


Figura 6.10: Montagem dos LEDs e do Alto Falante.

6.1.6 O *display* TFT

O display TFT 2.4"ILI9341 é um *shield*, logo, ele é projetado para ser encaixado direto no Arduino. Em um Arduino Uno, ele utiliza todos os pinos. Com a necessidade de utilizar pinos para a comunicação, o *display* foi conectado ao Arduino Mega, que contém mais pinos, e onde, ao invés de ser encaixado diretamente na placa, foi preferível utilizar *jumpers* para fazer a conexão, já que a tela ficará presa a um dos lados do cubo e longe do Arduino Mega.

A maneira como o *display* foi encaixado pode ser vista na Figura 6.12.

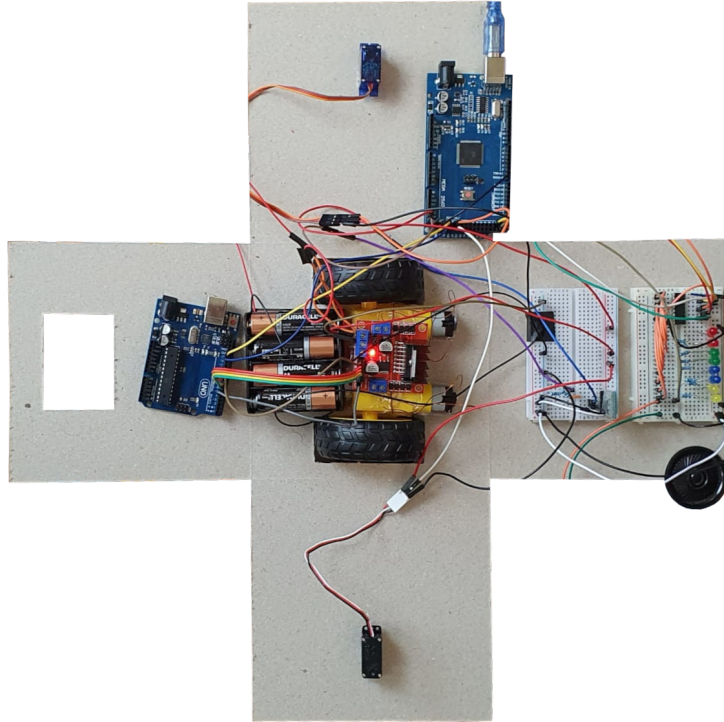


Figura 6.11: Montagem dos LEDs e alto falante no Cubo.

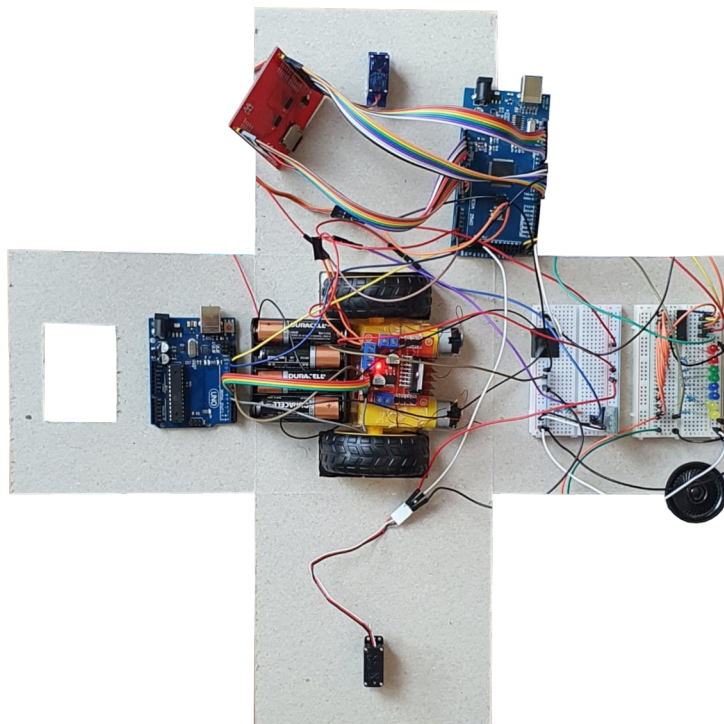


Figura 6.12: Montagem do *Display* TFT no Cubo.

6.1.7 Montagem Final

A montagem final do cubinho se deu com a adição das baterias que serão a fonte de energia dos Arduinos. Para o Arduino Uno foi utilizado uma bateria de 9V Alcalina, já no Arduino Mega foi necessário utilizar duas baterias 9V alcalinas em paralelo, já que ele alimenta praticamente todos os componentes do circuito, incluindo o *display* TFT, que utiliza uma boa quantidade de energia.

A montagem final do circuito pode ser visualizada com cubo aberto na Figura 6.13, com o cubo fechado dos lados e aberto em cima na Figura 6.14 e completamente fechado em Figura 6.15.

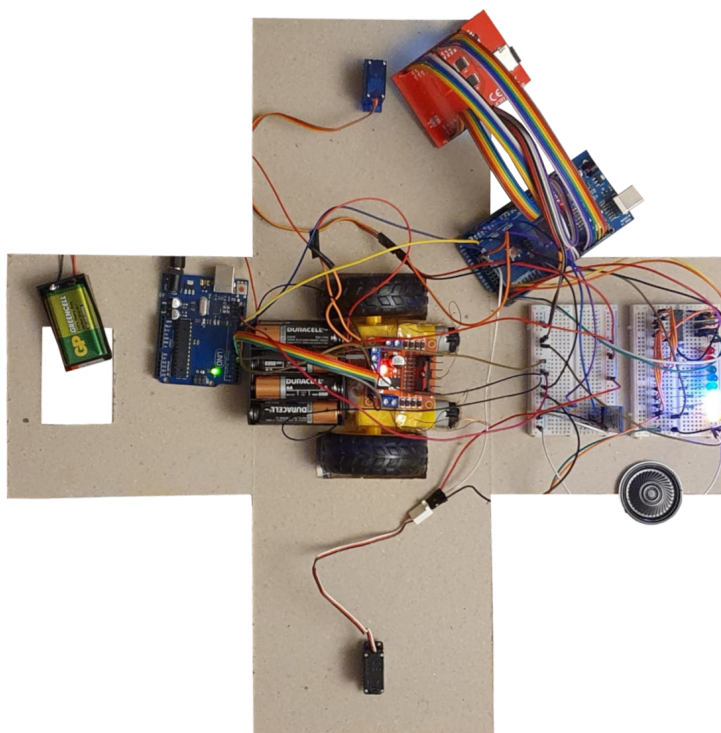


Figura 6.13: Montagem final do Cubinho.

6.2 O Cubinho

O objetivo do Cubinho é ser um reforçador para pessoas com TEA quando estão utilizando o aplicativo Prevenir. O espectro autista abrange uma variedade enorme de características, das mais diversas, fazendo com que seja praticamente impossível desenvolver um único reforço que seja efetivo para todas as pessoas com TEA. Tendo isso em mente, o projeto do *hardware* se deu levando em conta a necessidade de existir uma pluralidade de possibilidades de reforçadores, de vários tipos e intensidades diferentes.



Figura 6.14: Cubinho sem a parte de cima.

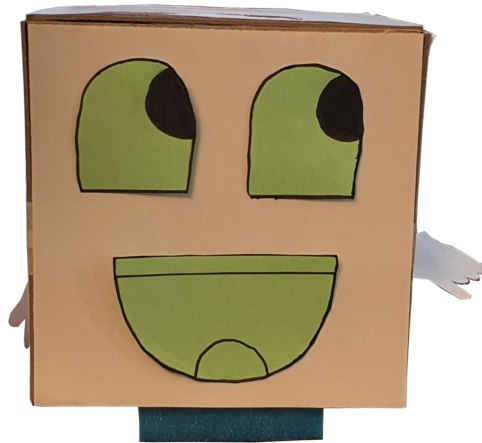


Figura 6.15: Cubinho montado.

A proposta do protótipo do projeto foi a seguinte: utilizar diferentes componentes eletrônicos em conjunto para criar um *hardware* que possua uma grande variedade de reforçadores e seja atrativo para pessoas com TEA tanto no aspecto visual quanto em qualquer outro sentido.

6.2.1 A Carcaça

A parte externa do *hardware* precisa ser o mais universal e customizável possível, afinal, em grande parte das vezes, será o aspecto visual a primeira característica notada, sendo de extrema importância que exista alguma simpatia pela aparência. Além disso, a fácil reprodução do *hardware* é essencial para seu sucesso, assim, não é possível escolher uma

carcaça de um artefato robótico ou brinquedo existente. Tendo esses aspectos em mente, um cubo foi selecionado para ser a parte exterior do *hardware*.

Um cubo é extremamente simples e pode ser feito com qualquer tipo de papel ou papelão. Para a confecção do protótipo foi utilizado um papel panamá de 2,5mm de espessura, que é forte o suficiente para aguentar o *hardware* e ainda assim não tão difícil de manusear. A base e os quatro lados foram cortados no papel, deixando o topo aberto, que será fechado após a inserção dos componentes eletrônicos. O papel cortado em forma de cubo pode ser visto na Figura 6.3.

Os componentes foram grudados no papel panamá utilizando fitas adesivas, cola e fitas dupla face.

A grande vantagem do cubo, além da facilidade de ser montado, é a grande possibilidade de customização. É possível desenhar ou imprimir imagens de animais, personagens ou qualquer outra figura que seja do agrado da pessoa com TEA para ser colada na superfície externa do cubo, gerando assim um alto grau de liberdade na hora de montar o Cubinho. Para esse projeto foram utilizados olhos e boca de papel para personalizar o cubo (Figura 6.16). Outras formas de como o Cubinho pode ser personalizado podem ser vistas na Figura 6.17 e na Figura 6.18.

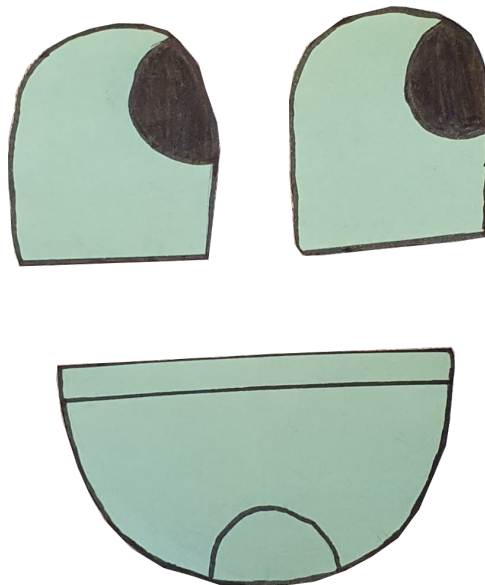


Figura 6.16: Imagem utilizada no rosto do Cubinho.

6.2.2 Os Reforçadores

Para a elaboração dos reforçadores, foram levados em conta a disponibilidade dos componentes, visando selecionar os de mais fácil acesso e que façam sentido com a proposta



Figura 6.17: Cubinho personalizado 1.

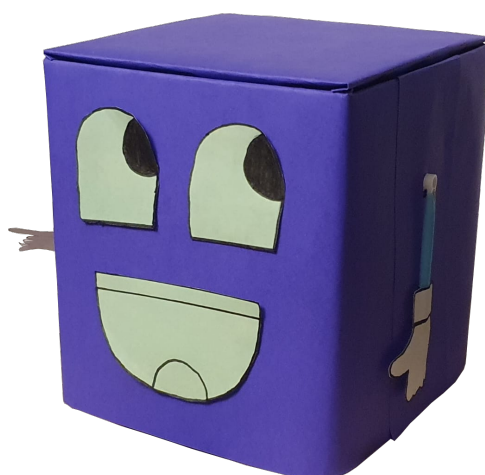


Figura 6.18: Cubinho personalizado 2.

do *hardware*. Alguns tipos de reforçadores foram estabelecidos, sendo eles:

- Movimento;
- Luz;
- Som;
- Imagens;

Para executar um reforço, o aplicativo Prevenir irá enviar ao Arduino um comando que define qual será o *feedback* desejado. A forma como o código do Arduino recebe os comandos e os executa pode ser visualizado na Figura 6.19.

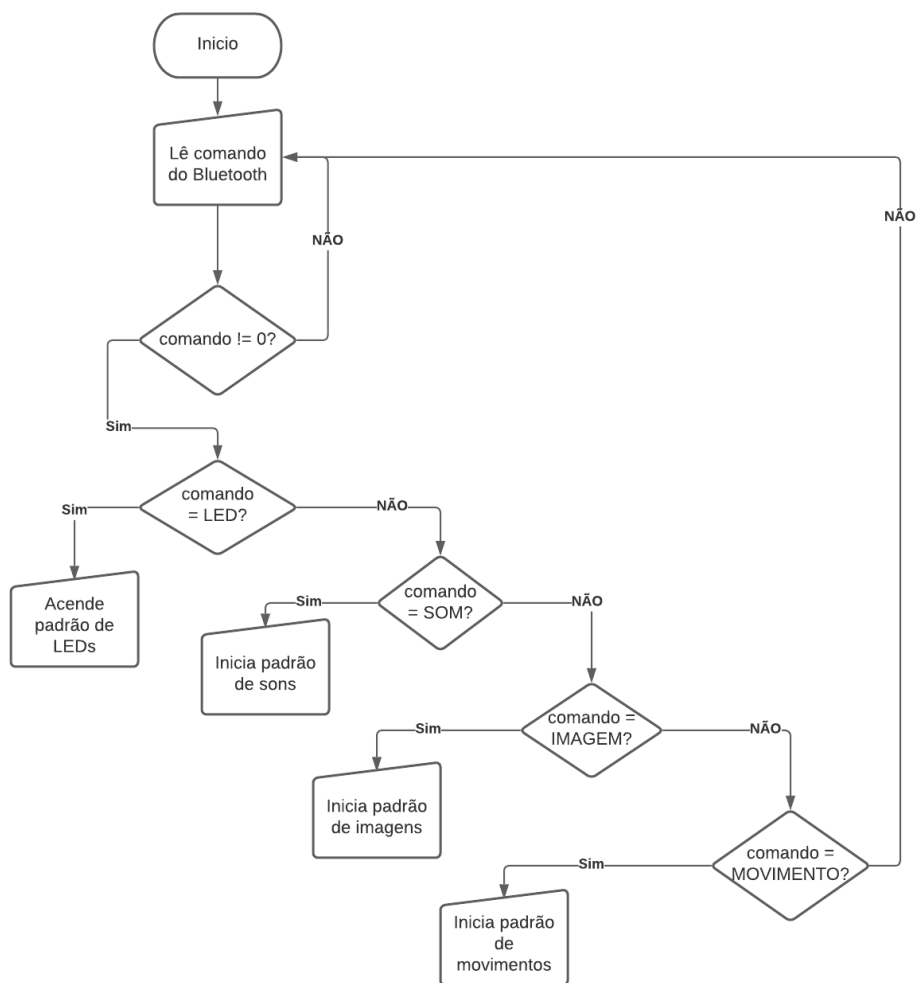


Figura 6.19: Diagrama de blocos do Código do Arduino.

Movimento

O primeiro reforçador foi o de movimento, o objetivo é que o cubo possa fazer movimentos que signifiquem aprovação ou reprovação diante da atitude tomada pela pessoa que está fazendo as atividades do aplicativo.

Foram adicionados dois motores DC com rodas ligados a uma ponte H na base do Cubo para executarem a movimentação. Dois motores servo foram adicionados, um em cada lado do cubo, com o intuito de simular braços, onde canudos foram acoplados com mãos desenhadas em papel nas extremidades: Uma mão com o polegar esticado e os outros dedos abaixados, fazendo o sinal de “certo”, que pode ser visto na Figura 6.21 e a outra com o indicador esticado e os outros dedos abaixados, que pode ser visto na Figura 6.22. Os braços ficam paralelos no corpo quando não estão em movimento.

Foram desenvolvidos então 3 movimentos:

- Levantar o braço que possui a mão com o polegar levantado e fazer um movimento “positivo”.
- Levantar o braço que possui o dedo indicador para cima e rotacionar o corpo no próprio eixo de um lado para o outro fazendo um sinal de “negativo”.
- Fazer uma rotação completa no próprio eixo.

Para realizar o movimento de “positivo”, foi necessário rotacionar o motor servo acoplado à mão com polegar para cima. É realizada uma rotação para cima até uma altura que pareça condizente para o movimento desejado, voltando um pouco para baixo em seguida, depois para cima e por último voltando para a posição inicial, simulando um movimento de aprovação.

No movimento de “negativo”, primeiramente o motor servo acoplado ao braço com indicador levantado é rotacionado até uma certa altura. Em seguida, os motores DC das rodas são girados cada um em uma direção para que o cubo gire no seu próprio eixo, fazendo um ângulo de aproximadamente 60 graus para um lado e depois voltando a posição inicial. Esse movimento se repete duas vezes, com a mão sempre levantada. Ao final do movimento, a mão volta para a posição inicial.

Para realizar o movimento de rotação, cada roda gira em uma direção por determinado tempo, até finalizar a volta no seu eixo.

Com o objetivo de estabilizar o cubo em pé, foram utilizados pedaços de esponja cortados na base do cubo. Quando ele se movimenta, elas dão suporte para frente e para trás sem que o cubo fique instável. (Figura 6.20).

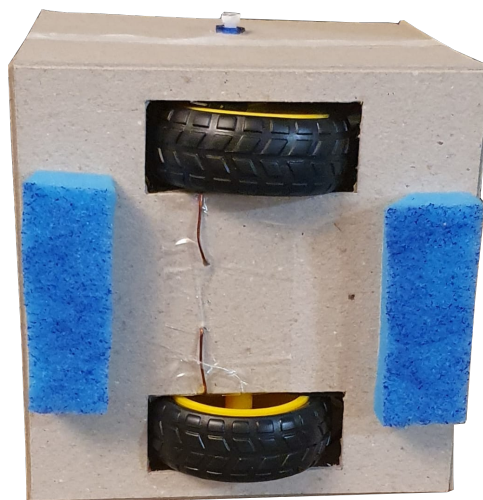


Figura 6.20: Parte de baixo do Cubinho.



Figura 6.21: Mão com polegar do Cubinho.

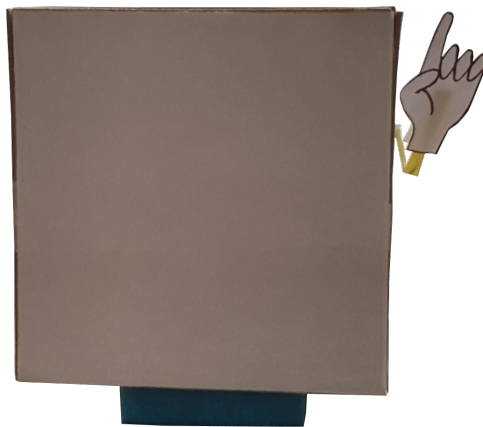


Figura 6.22: Mão com indicador do Cubinho.

Luz

O reforçador de luz é bem simples. O intuito era disponibilizar uma certa variedade de cores possíveis para serem acionadas individualmente além de alguns modos que utilizam várias luzes, caso seja necessário um estímulo mais forte. Foram utilizados então 8 LEDs ao total, sendo 2 vermelhos, 2 verdes, 2 azuis e 2 amarelos.

As ações que os LEDs executam são:

- Acender uma cor(vermelho, verde, azul ou amarelo) durante meio segundo.
- Piscar todos os 8 LEDs 4 vezes esperando 0,2 segundos entre piscadas.
- Acender os LEDs em sequência dois de cada vez, apagando o primeiro e acendendo o seguinte de forma a fazer um efeito de “trem” que se movimenta em uma direção. O movimento de vai e vem é executado 3 vezes.

Os LEDs ficaram dispostos na parte de cima do cubo, onde foi recortado um retângulo que foi coberto com papel manteiga, onde a luminosidade dos LEDs pode ser vista(Figura 6.23).

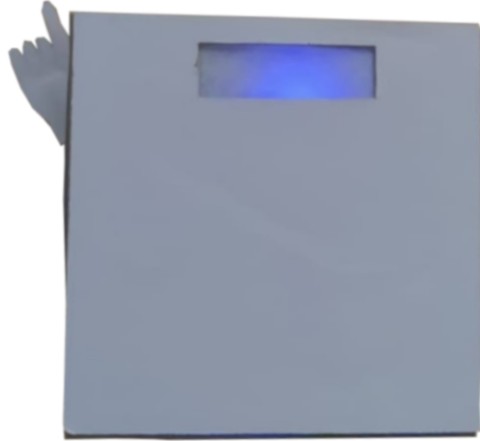


Figura 6.23: Parte de cima do Cubinho com os LEDs.

Som

Um alto falante foi inserido no cubo para servir como o reforçador sonoro. Existem diversos sons que podem ser reproduzidos, desde notas individuais que variam de agudas a graves, até pequenas melodias e sons mais complexos. O alto falante pode ser visto no cubo na Figura 6.24.

Imagem

Um reforçador interessante adicionado no Cubinho foi o de reprodução de imagem. Um *display* TFT exibe imagens motivacionais que podem ser, por exemplo, um sorvete, ursinho de pelúcia, pirulito, bicicleta, chocolate, bola entre outros.

Essas imagens estão no cartão micro SD inserido do *display* TFT e a forma como elas aparecem no cubo pode ser visto na Figura 6.25.

6.3 O Aplicativo Prevenir

O objetivo final do Cubinho é ser utilizado em conjunto com o aplicativo para *smartphones* Prevenir. Para conectar o Cubinho com o aplicativo, é necessário que o celular possua *bluetooth*.

O aplicativo Prevenir é composto de 3 atividades que possuem como objetivo instruir pessoas com TEA sobre objetos domésticos perigosos e como se portar diante de uma

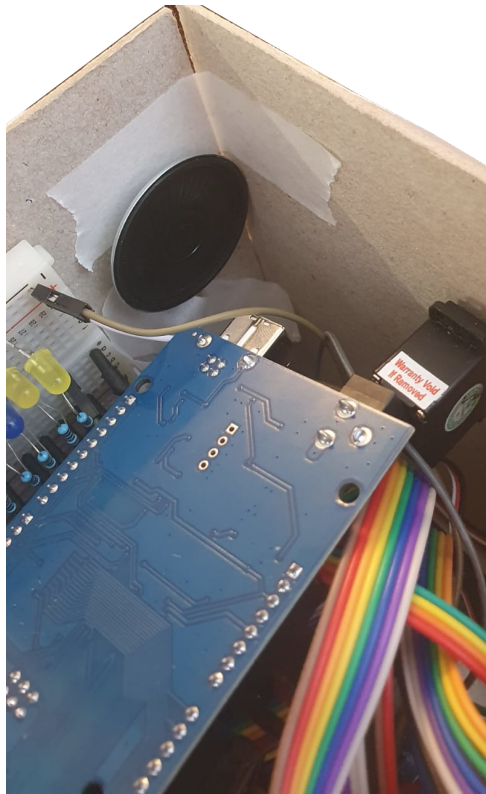


Figura 6.24: Alto falante grudado no Cubinho.



Figura 6.25: *Display* TFT no Cubinho.

possível situação de risco. A primeira atividade visa ensinar quais objetos não são seguros e que não se deve tocar neles (Figura 6.26). A segunda e a terceira atividade visam ensinar a pessoa a diferenciar objetos seguros de não seguros (Figura 6.27). A integração do Cubinho com o aplicativo é realizada por meio da ativação de algum dos reforços

quando uma atividade é acertada ou errada, dependendo de como foi configurado.



Figura 6.26: Prevenir Atividade 1.



Figura 6.27: Prevenir Atividade 2.

Dentro do aplicativo foi inserida uma seção de configurações, onde está disponível a opção “Configurar Cubinho”. Nessa seção é possível fazer a conexão com o *hardware* através da opção de “Configurar Bluetooth” e configurar os reforçadores na opção “Configurar Feedback”. No caso do Cubinho, o nome do *bluetooth* é HC-05, o nome do módulo utilizado. A tela de configurações pode ser vista na Figura 6.28, a de configuração do Cubinho na Figura 6.29 e a de configuração do *bluetooth* na Figura 6.30.

Após estar conectado, o próximo passo é selecionar quais serão os reforços que o Cubinho irá executar no caso de acerto ou erro da atividade. A escolha dos reforços é opcional, sendo que o reforço de erro pode ser utilizado para inibir algum tipo de comportamento inadequado. Essas configurações devem ser baseadas na realidade de cada indivíduo e é possível testar cada *feedback* ao selecioná-lo. A tela de configuração dos reforços pode ser visualizada na Figura 6.31 e a tela de seleção de reforços pode ser visualizada na Figura 6.32.

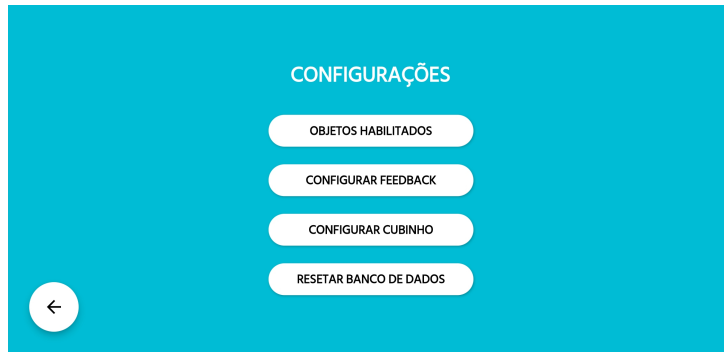


Figura 6.28: Tela de configurações do aplicativo Preventir.

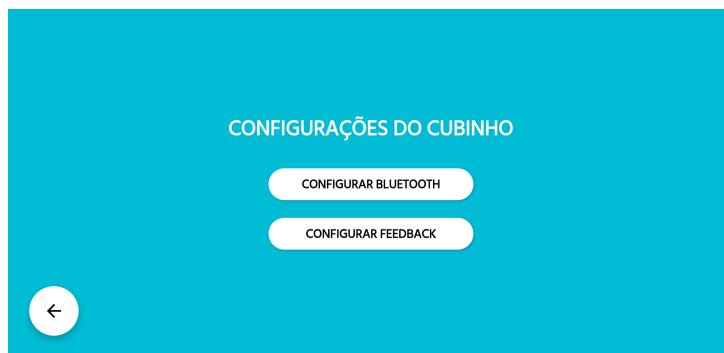


Figura 6.29: Tela de configuração do Cubinho no aplicativo Preventir.



Figura 6.30: Tela de configuração do *bluetooth* no aplicativo Preventir.

Após a escolha dos reforços, o aplicativo pode ser utilizado normalmente. Sempre que alguma atividade for acertada, o reforço de acerto ocorrerá, e sempre que houver algum erro, o reforço de erro ocorrerá.

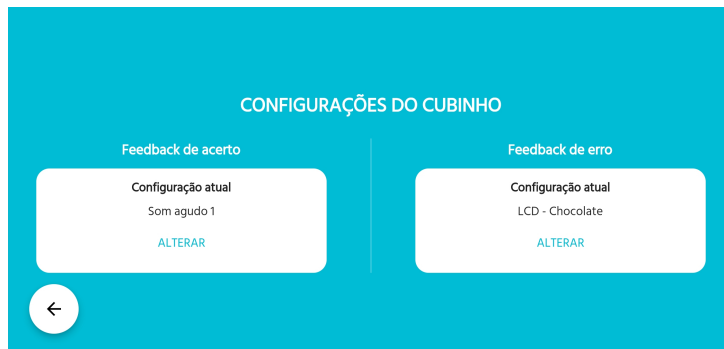


Figura 6.31: Tela de configurações dos reforços no aplicativo Prevenir.

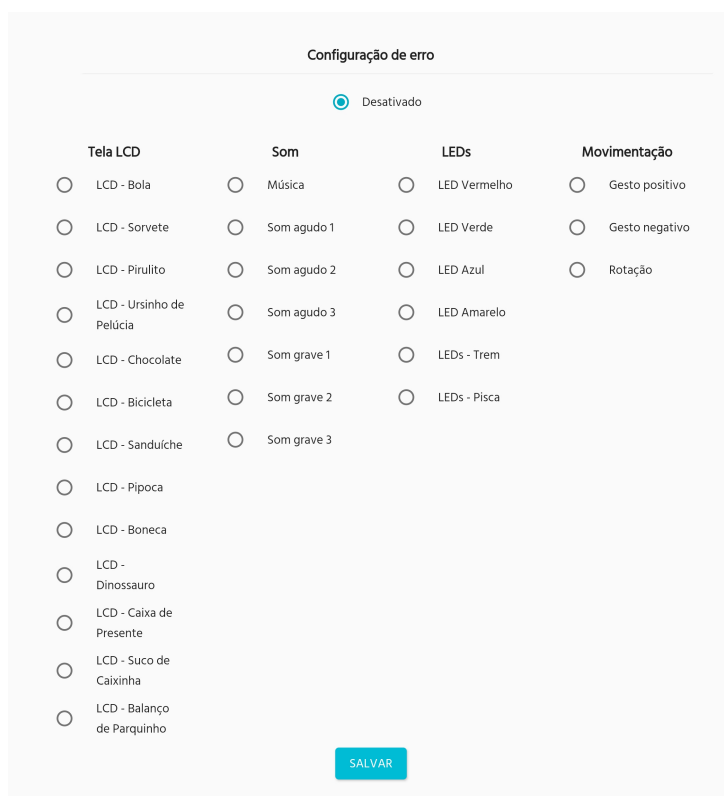


Figura 6.32: Tela de seleção dos reforços no aplicativo Prevenir.

Capítulo 7

Conclusão

Este trabalho descreve o desenvolvimento de um artefato robótico em forma de cubo, o Cubinho, que possui diversas funcionalidades para motivar certos comportamentos em pessoas com Transtorno do Espectro Autista enquanto utilizam o aplicativo Prevenir, voltado para a prevenção de acidentes domésticos para pessoas com TEA. O Cubinho pode acender luzes de diferentes cores, emitir sons de diferentes frequências e intensidades, exibir imagens e se movimentar, tudo de maneira personalizável às necessidades do usuário, já que não existe um único jeito de motivar todas as pessoas com TEA.

Dentro do Cubinho existe um Arduino Mega, Arduino Uno, *display* TFT, módulo *bluetooth*, LEDs, alto falante, motores servo e motores DC, que juntos oferecem uma grande variedade de reforçadores. Essa quantidade de componentes é fundamental para o sucesso do projeto, permitindo um maior alcance do *hardware*.

A forma como a integração dos componentes eletrônicos foi facilmente executada e como os testes dos componentes foram bem sucedidos mostra que o Arduino foi uma boa escolha como placa controladora do *hardware*, permitindo que o Cubinho seja facilmente reproduzido.

Dentre as dificuldades de se realizar esse projeto a maior está no entendimento e interpretação das necessidades e dificuldades de pessoas com TEA e na transformação dessas informações em um produto concreto, utilizando componentes eletrônicos para atender essas necessidades e tentar de alguma maneira trazer mais independência e segurança para a vida de pessoas com Transtorno do Espectro Autista.

Uma importante adição futura para esse trabalho seria a realização de testes do *hardware* com pessoas com TEA, o que não foi possível devido às medidas de distanciamento social trazidas pela atual pandemia do Covid-19, com o intuito de validar na prática os conceitos teóricos utilizados e, posteriormente, efetuar possíveis correções utilizando o *feedback* observado durante os testes. A inclusão de novos reforçadores ou de novas variações dos existentes também seriam modificações bem-vindas.

Referências

- [1] Rabello, Clara Senra: *Prevenir – aplicativo educacional para smartphones de apoio à prevenção de acidentes domésticos para pessoas com autismo*. TCC (graduação) – Engenharia de Computação, Universidade de Brasília, 2021. v
- [2] Team, Arduino: *Pwm*. <https://www.arduino.cc/en/Tutorial/Foundations/PWM>. ix, 20, 21
- [3] Team, Arduino: *Arduino mega*. <https://store.arduino.cc/usa/mega-2560-r3>. ix, 19, 24, 25, 26
- [4] Team, Arduino: *Arduino uno rev3*. <https://store.arduino.cc/usa/arduino-uno-rev3>. ix, 19, 24, 27, 28
- [5] Team, Arduino: *Fade*. <https://www.arduino.cc/en/Tutorial/BuiltInExamples/Fade>. ix, 33, 34
- [6] *Sweep*. <https://www.arduino.cc/en/Tutorial/LibraryExamples/Sweep>. ix, 35, 36
- [7] Instruments, Texas: *74hc595 datasheet*. <https://www.ti.com/lit/ds/symlink/cd74hc595.pdf>. x, 49
- [8] Comportamento, Departamento Científico de Pediatria do Desenvolvimento e: *Manual de orientação transtorno do espectro autista*. 2019. https://www.sbp.com.br/fileadmin/user_upload/Ped._Desenvolvimento_-_21775b-MO_-_Transtorno_do_Espectro_do_Autismo.pdf. 1, 2, 6
- [9] Caminha, Vera Lúcia Prudência dos Santos, Julliane Yoneda Huguenin, Lúcia Maria de Assis e Priscila Pires Alves: *Autismo: vivências e Caminhos*. BLUCHER, SP, BR, 2016. 1
- [10] DEN BROK, Wilco L. J. E. e Paula STERKENBURG: *Self-controlled technologies to support skill attainment in persons with an autism spectrum disorder and/or an intellectual disability: a systematic literature review, disability and rehabilitation: Assistive technology*. Disability and rehabilitation. Assistive technology, 2014. 1
- [11] ABA, Terapia: *Por que usamos reforçadores?* <https://www.terapiaaba.com.br/noticia/Por-que-usamos-reforcadores/32#:~:text=Um%20refor%C3%A7ador%2C%20de%20modo%20simplificado,a%20um%20comportamento%20%E2%80%9Cadequado%E2%80%9D.&text=Quando%20uma%20crian%C3%A7a%20neurot%C3%ADpica%20fala,refor%C3%A7ando%20o%20comportamento%20de%20falar>. 2, 8

- [12] Infância, Rede Nacional Primeira: *Plano nacional primeira infância: 2010 - 2022 | 2020 - 2030*. ANDI Comunicação e Direitos, páginas 164–172, 2020. 2
- [13] BORBA, M. M. C. e R. S. BARROS: *Ele é autista: como posso ajudar na intervenção? um guia para profissionais e pais com crianças sob intervenção analítico-comportamental ao autismo*. Cartilha da Associação Brasileira de Psicologia e Medicina Comportamental (ABPMC), 2018. 2
- [14] MARFINATI, Anahi Canguçu e Jorge Luís Ferreira ABRAO: *Um percurso pela psiquiatria infantil: dos antecedentes históricos à origem do conceito de autismo*. Estilos da clinica. [online]., 19(02):244–262, 2014. <https://dx.doi.org/10.11606/issn.1981-1624.v19i2p244-262>. 5, 6
- [15] KANNER, Leo: *Autistic disturbances of affective contact*. Nervous Child, 02:217–250, 1943. 5
- [16] ASPERGER, Hans: *Autistic psychopathy in childhood*. Cambridge: Cambridge University Press, páginas 37–92, 1991. 5
- [17] WING, Lorna e Judith GOULD: *Severe impairments of social interaction and associated abnormalities in children: Epidemiology and classification*. Journal of Autism and Developmental Disorders, 02:11–29, 1979. 5
- [18] (APA), American Psychiatric Association: *Manual Diagnóstico e Estatístico de Transtornos Mentais DSM-5*. Artmed Editora, 2014. 6
- [19] Saúde, Organização Mundial da: *Cid 11*. <https://www.who.int/classifications/classification-of-diseases>. 6
- [20] MELLO, Ana Maria S. Ros de: *Autismo : guia prático*. Corde, 2007. 7, 8, 9
- [21] Autista, Associação de Amigos do: *Diagnóstico*. <https://www.ama.org.br/site/autismo/diagnostico/>. 7
- [22] Consultants, Pyramid Educational: *Sistema de comunicação por troca de figuras (pecs)*. <https://pecs-brazil.com/sistema-de-comunicacao-por-troca-de-figuras-pecs/>. 8
- [23] Amorim, Letícia Calmon Drummond: *Tratamento*. <https://www.ama.org.br/site/autismo/tratamento/>. 8
- [24] Brasil, Presidência da República do: *Constituição da república federativa do brasil de 1988*. http://www.planalto.gov.br/ccivil_03/constituicao/constituicao.htm. 9
- [25] Brasil, Presidência da República do: *Lei no 12.764, de 27 de dezembro de 2012*. http://www.planalto.gov.br/ccivil_03/ato2011-2014/2012/lei/l12764.htm. 9
- [26] Brasil, Presidência da República do: *Decreto nº 6.949, de 25 de agosto de 2009*. http://www.planalto.gov.br/ccivil_03/ato2007-2010/2009/decreto/d6949.htm#:text=1.,todos%20os%20aspectos%20da%20vida.9

- [27] Brasil, Presidência da República do: *Lei no 13.146, de 6 de julho de 2015*. http://www.planalto.gov.br/ccivil_03/ato2015-2018/2015/lei/l13146.htm. 9, 10
- [28] Brasil, Presidência da República do: *Lei no 10.741, de 1º de outubro de 2003*. http://www.planalto.gov.br/ccivil_03/leis/2003/l10.741.htm. 9
- [29] Senado, Agência: *Sancionada lei que inclui conceito de educação e aprendizagem ao longo da vida na ldb*, 2018. <https://www12.senado.leg.br/noticias/materias/2018/03/07/sancionada-lei-que-inclui-conceito-de-educacao-e-aprendizagem-ao-longo-da-vida-na-ldb>. 9
- [30] Brasil, Presidência da República do: *Lei nº 8.742, de 7 de dezembro de 1993*. http://www.planalto.gov.br/ccivil_03/leis/l8742.htm. 10
- [31] Brasil, Presidência da República do: *Lei nº 9.394, de 20 de dezembro de 1996*. http://www.planalto.gov.br/ccivil_03/leis/l9394.htm. 10
- [32] Brasil, Presidência da República do: *Decreto nº 7.611, de 17 de novembro de 2011*. http://www.planalto.gov.br/ccivil_03/ato2011-2014/2011/decreto/d7611.htm. 10
- [33] Brasil, Presidência da República do: *Lei nº 7.853, de 24 de outubro de 1989*. http://www.planalto.gov.br/ccivil_03/leis/l7853.htm. 10
- [34] Brasil, Presidência da República do: *Lei nº 8.899, de 29 de junho de 1994*. http://www.planalto.gov.br/ccivil_03/leis/l8899.htm#text : : *text = LEI%20N%C2%BA%208.899%2C%20DE%2029%20DE%20JUNHO%20DE%201994.text = Concede%20passe%20livre%20C3%A0s%20pessoas, eu%20sanciono%20a%20seguinte%20lei%3Ate3%C2%BA%20Esta%20lei%20entra%20em%20vigor%20na%20data%20de%20sua%20publica%C3%A0*
- [35] Brasil, Presidência da República do: *Lei no 10.048, de 8 de novembro de 2000*. http://www.planalto.gov.br/ccivil_03/leis/l10048.htm. 10
- [36] DEFICIÊNCIA, COORDENADORIA NACIONAL PARA INTEGRAÇÃO DA PESSOA PORTADORA DE: *Ata vii reunião do comitê de ajudas técnicas - cat corde / sedh / pr realizada nos dias 13 e 14 de dezembro de 2007*. https://www.assistiva.com.br/Ata_VII_Reuni%C3%A3o_do_Comite_de_Ajudas_T%C3%A9cnicas.pdf. 10
- [37] BERSCH, Rita: *Introdução À tecnologia assistiva*, 2017. https://www.assistiva.com.br/Introducao_Tecnologia_Assistiva.pdf. 10
- [38] Brasil, W3C: *Cartilha acessibilidade na web w3c brasil*. <https://ceweb.br/media/docs/publicacoes/1/cartilha-w3cbr-acessibilidade-web-fasciculo-I.pdf>. 12
- [39] Governo Eletrônico, Departamento de: *emag - modelo de acessibilidade em governo eletrônico*. <http://emag.governoeletronico.gov.br/>, 2014. 12
- [40] BRITTO, Talita Cristina Pagani: *“gaia: Uma proposta de guia de recomendações de acessibilidade web com foco em aspectos do autismo”*. <https://repositorio.ufscar.br/bitstream/handle/ufscar/8683/DissTCPB.pdf?sequence=1&isAllowed=y>, 2016. 13

- [41] BRITTO, Talita Cristina Pagani: *Sites inclusivos a pessoas com autismo - gaia*. <https://gaia.wiki.br/>. 13
- [42] MAGATON, Heloise Cristini e Silvia Amélia BIM: *Recomendações para o desenvolvimento de softwares voltados para crianças com transtorno do espectro autista*. <https://www.br-ie.org/pub/index.php/rbie/article/view/v27n02112139/6049>, 2018. 13
- [43] Team, Arduino: *About us - arduino*. <https://www.arduino.cc/en/Main/AboutUs>. 14
- [44] Team, Arduino: *Introduction - arduino*. <https://www.arduino.cc/en/Guide/Introduction>. 14, 15, 18
- [45] Team, Arduino: *Environment - arduino*. <https://www.arduino.cc/en/Guide/Environment>. 15, 16
- [46] Team, Processing: *Overview. a short introduction to the processing software and projects from the community*. <https://processing.org/overview/>. 15
- [47] Team, Arduino: *Libraries - arduino*. <https://www.arduino.cc/en/reference/libraries>. 16
- [48] Team, Arduino: *Glossary - arduino*. <https://www.arduino.cc/glossary/en/>. 17
- [49] BARRAGÁN, Hernando: *The untold history of arduino*. <https://arduinhistory.github.io/>. 17
- [50] Team, Circuito: *Everything you need to know about arduino code*, 2018. <https://www.circuito.io/blog/arduino-code/#:~:text=What%20language%20is%20Arduino%3F,and%20compiled%20to%20machine%20language>. 17
- [51] Team, Arduino: *Language reference - arduino*. <https://www.arduino.cc/reference/en/>. 17
- [52] Team, Arduino: *pinmode()*. <https://www.arduino.cc/reference/en/language/functions/digital-io/pinmode/>. 17
- [53] Team, Arduino: *digitalread()*. <https://www.arduino.cc/reference/en/language/functions/digital-io/digitalread/>. 17
- [54] Team, Arduino: *digitalwrite()*. <https://www.arduino.cc/reference/en/language/functions/digital-io/digitalwrite/>. 17
- [55] Team, Arduino: *analogread()*. <https://www.arduino.cc/reference/en/language/functions/analog-io/analogread/>. 18
- [56] Team, Arduino: *analogwrite()*. 16-<https://www.arduino.cc/reference/en/language/functions/analog-io/analogwrite/>. 18
- [57] Team, Arduino: *Arduino nano*. <https://store.arduino.cc/usa/arduino-nano>. 19

- [58] KEIM, Robert: *What is a microcontroller? the defining characteristics and architecture of a common component*, 2019. <https://www.allaboutcircuits.com/technical-articles/what-is-a-microcontroller-introduction-component-characteristics-component/>. 19
- [59] Team, Arduino: *Digital pins*. <https://www.arduino.cc/en/Tutorial/Foundations/DigitalPins>. 20
- [60] Team, Arduino: *Serial*. <https://www.arduino.cc/reference/pt/language/functions/communication/serial/>. 20
- [61] Team, Arduino: *Analog input pins*. <https://www.arduino.cc/en/Tutorial/Foundations/AnalogInputPins>. 21
- [62] ST: *L298n datasheet*. <https://html.alldatasheet.com/html-pdf/22440/STMICROELECTRONICS/L298N/1619/1/L298N.html>. 24
- [63] Automation, OMRON Industrial: *Technical explanation for servomotors and servo drives*. https://www.ia.omron.com/data_pdf/guide/14/servo_tg_e_1_1.pdf. 26
- [64] HELERBROCK, Rafael: *O que é led?* <https://brasilecola.uol.com.br/o-que-e/fisica/o-que-e-led.htm>. 28
- [65] Britannica, The Editors of Encyclopaedia: *"loudspeaker"*, 2020. <https://www.britannica.com/technology/loudspeaker>. 31
- [66] ILITEK: *a-si tft lcd single chip driver 240rgbx320 resolution and 262k color specification*. <https://cdn-shop.adafruit.com/datasheets/ILI9341.pdf>. 32
- [67] Team, Adafruit: *Tftlcd library*. <https://github.com/adafruit/TFTLCD-Library>. 39