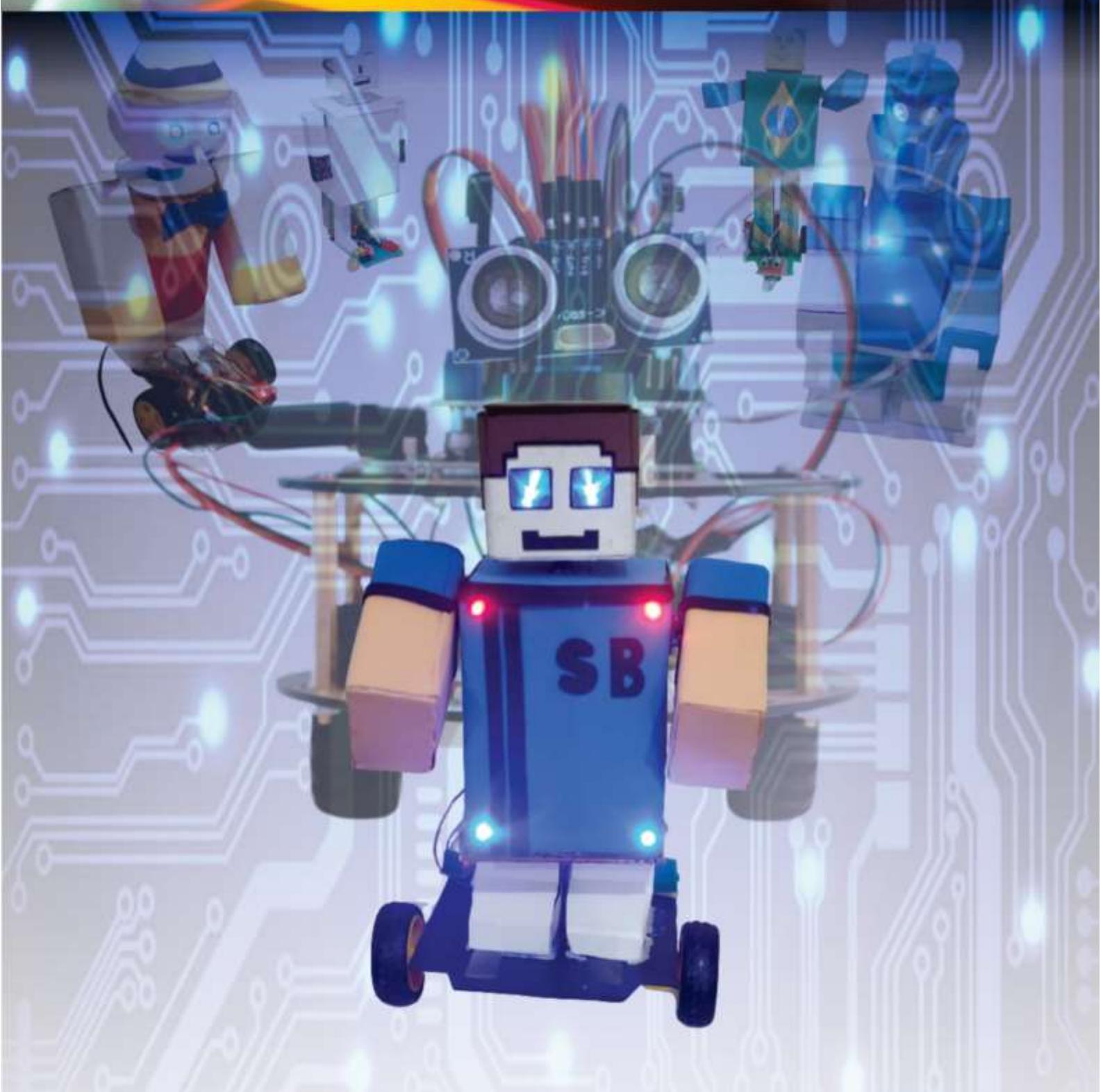


ROBÓTICA

&

A MÉTRICA DAS FORMAS



DENIS HEITOR DAMASCENO DA SILVA
DR. CARLOS ALBERTO DE MIRANDA PINHEIRO
DRA. CINTHIA CUNHA MARADEI PEREIRA

REITOR DA UEPA
Clay Anderson Nunes Chagas

PRÓ-REITOR DE PESQUISA E PÓS-GRADUAÇÃO
Jofre Jacob da Silva Freitas

DIRETOR DO CCSE
Anderson Madson de Oliveira Maia

COORDENADOR DO PROGRAMA
Pedro Franco de Sá

VICE-COORDENADORA DO PROGRAMA
Ana Kely Martins Silva

Diagramação e Capa: Os Autores

Revisão: Os Autores

Conselho Editorial

Profa. Dra. Acylena Coelho Costa
Profa. Dra. Ana Kely Martins da Silva
Prof. Dr. Antonio José Lopes
Prof. Dr. Admilson Alcântara da Silva
Prof. Dr. Benedito Fialho Machado
Prof. Dr. Carlos A. Raposo da Cunha
Prof. Dr. Carlos A. de Miranda Pinheiro
Profa. Dra. Celsa H. de M. Maranhão
Profa. Dra. Cinthia C. Maradei Pereira
Profa. Dra. Claudianny A. Noronha
Profa. Dra. Cristina Lúcia Dias Vaz
Prof. Dr. Dorival Lobato Junior
Prof. Dr. Dúcival Carvalho Pereira
Profa. Dra. Eliza Souza da Silva
Prof. Dr. Fábio José da Costa Alves
Prof. Dr. Francisco Hermes S. da Silva
Prof. Dr. Geraldo Mendes de Araújo
Profa. Dra. Glaudianny A. Noronha
Prof. Dr. Gustavo Nogueira Dias

Prof. Dr. Heliton Ribeiro Tavares
Prof. Dr. João Cláudio B. Quaresma
Prof. Dr. José Antonio Oliveira Aquino
Prof. Dr. José Augusto N. Fernandes
Prof. Dr. José Messildo Viana Nunes
Prof. Dr. Márcio Lima do Nascimento
Prof. Dr. Marcos Antônio F. de Araújo
Prof. Dr. Marcos Monteiro Diniz
Profa. Dra. Maria de Lourdes S. Santos
Profa. Dra. Maria Lúcia P. C. Rocha
Prof. Dr. Miguel Chaquiam
Prof. Dr. Natanael Freitas Cabral
Prof. Dr. Pedro Franco de Sá
Prof. Dr. Raimundo O. M. Figueiredo
Profa. Dra. Rita Sidmar Alencar Gil
Prof. Dr. Roberto Paulo Bibas Fialho
Profa. Dra. Talita C. da S. de Almeida

Comitê de Avaliação

Cinthia C. Maradei Pereira
Carlos A. de Miranda Pinheiro
Acylena Coelho Costa
Amari Goulart

SILVA, Denis Heitor D. da; PEREIRA, Cinthia C. Maradei; PINHEIRO, Carlos A. Miranda. ROBÓTICA & A MÉTRICA DAS FORMAS. Produto Educacional do Programa de Pós-Graduação em Ensino de Matemática, Curso de Mestrado Profissional em Ensino de Matemática da Universidade do Estado do Pará, (PPGEM/UEPA), 2023.

ISBN: 978-65-5291-015-8

Ensino de Prismas Retos: Reconhecimento e Área das Superfícies Planas; Robótica Educacional; Scratch for Arduino – S4A.

Promoção

**Grupo de Pesquisa Cognição e Educação Matemática da Universidade do
Estado do Pará**

Autores

Denis Heitor Damasceno da Silva

Dr. Carlos Alberto de Miranda Pinheiro

Dra. Cinthia Cunha Maradei Pereira



UNIVERSIDADE DO ESTADO DO PARÁ

**PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO CENTRO DE
CIÊNCIAS SOCIAIS E EDUCAÇÃO**

PROGRAMA DE MESTRADO PROFISSIONAL EM ENSINO DE MATEMÁTICA

FICHA DE AVALIAÇÃO DE PRODUTOS EDUCACIONAIS – BANCA EXAMINADORA

Título: “**ROBÓTICA E A MÉTRICA DAS FORMAS**”.

Mestrando: **DENIS HEITOR DAMASCENO DA SILVA**

Data da avaliação: **12/12/2023**

PÚBLICO ALVO DO PRODUTO EDUCACIONAL

a) *Destinado à:*

- () Estudantes do Ensino Fundamental () Estudantes do Ensino Médio
() Professores do Ensino Fundamental (X) Professores do Ensino Médio ()

Outros: _____

INFORMAÇÕES SOBRE O PRODUTO EDUCACIONAL

a) *Tipo de Produto Educacional*

- (X) Sequência Didática () Página na Internet () Vídeo ()
Texto Didático (alunos/professores) () Jogo Didático () Aplicativo
() Software () Outro: _____

b) *Possui URL:* () Sim, qual o URL: _____

- () Não () Não se aplica

c) *É coerente com a questão-foco da pesquisa?*

(x) Sim

() Não. Justifique? _____

d) *É adequado ao nível de ensino proposto?*

(x) Sim

() Não. Justifique? _____

e) *Está em consonância com a linguagem matemática do nível de ensino proposto?*

(x) Sim

() Não. Justifique? _____

ESTRUTURA DO PRODUTO EDUCACIONAL

- a) *Possui sumário:* (x) Sim () Não () Não se aplica
b) *Possui orientações ao professor:* (x) Sim () Não () Não se aplica
c) *Possui orientações ao estudante:* () Sim () Não (x) Não se aplica
d) *Possui objetivos/finalidades:* (x) Sim () Não () Não se aplica
e) *Possui referências:* (x) Sim () Não () Não se aplica
f) *Tamanho da letra acessível:* (x) Sim () Não () Não se aplica
g) *Ilustrações são adequadas:* (x) Sim () Não () Não se aplica

CONTEXTO DE APLICAÇÃO DO PRODUTO EDUCACIONAL

a) *Foi aplicado?*

(X) Sim, onde: Escolas estaduais de ensino médio

() Não, justifique: _____

() Não se aplica

b) *Pode ser aplicado em outros contextos de Ensino?*

(X) Sim, onde: Curso de graduação de matemática

() Não, justifique: _____

() Não se aplica

c) *O produto educacional foi validado antes de sua aplicação?*

(x) Sim, onde: Escola

() Não, justifique: _____

() Não se aplica

d) *Em qual condição o produto educacional foi aplicado?*

(x) na escola, como atividade regular de sala de aula ()

na escola, como um curso extra

() outro: _____

e) *A aplicação do produto envolveu (marque as alternativas possíveis):*

() Alunos do Ensino Fundamental (x)

Alunos do Ensino Médio

() Professores do Ensino Fundamental (x)

Professores do Ensino Médio

() outros membros da comunidade escolar, tais como _____

() outros membros da comunidade, tais como _____

O produto educacional foi considerado:

(x) APROVADO

() APROVADO COM MODIFICAÇÕES

() REPROVADO

MEMBROS DA BANCA

Profa. Dra. Cinthia Cunha Maradei Pereira (Presidente)

Doutora em Bioinformática

IES de obtenção do título: UFPA

Prof. Dr. Carlos Alberto de Miranda Pinheiro (Coorientador)

Doutor em Educação Matemática

IES de obtenção do título: PUC/SP

Profa. Dra. Acylene Coelho Costa (Examinador 01)

Doutor em Educação Matemática

IES de obtenção do título: PUC/SP

Prof. Dr. Amari Goulart (Examinador 02)

Doutor em Educação Matemática

IES de obtenção do título: PUC/SP

Assinaturas

Documento assinado digitalmente



CINTHIA CUNHA MARADEI PEREIRA

Data: 18/12/2023 20:58:42-0300

Verifique em <https://validar.itd.gov.br>

Documento assinado digitalmente



CARLOS ALBERTO DE MIRANDA PINHEIRO

Data: 09/01/2024 09:59:08-0300

Verifique em <https://validar.itd.gov.br>

Documento assinado digitalmente



ACYLENA COELHO COSTA

Data: 19/12/2023 22:57:11-0300

Verifique em <https://validar.itd.gov.br>

Documento assinado digitalmente



AMARI GOULART

Data: 19/12/2023 10:31:44-0300

Verifique em <https://validar.itd.gov.br>

SUMÁRIO

1 - APRESENTAÇÃO	6
2 - SCRATCH, ARDUINO E SCRATCH FOR ARDUINO – S4A.....	6
3 - A ROBÓTICA EDUCACIONAL	11
4 - PROBLEMA PROPOSTO.....	12
4.1 - Atividade I.....	13
4.2 - Atividade II.....	14
4.3 - Atividade III.....	15
4.4 - Atividade IV	16
5 - CONSTRUÇÃO E PROGRAMAÇÃO DO PROTÓTIPO ROBOTIZADO.....	16
5.1 - Cabeça Robótica.....	16
5.2 - Braços Robóticos	20
5.3 - Tronco Robótico	24
5.3.1 - Circuito Elétrico.....	25
5.4 – Pernas Robóticas	28
5.4.1 – Base Direcional	29
6 – JUNÇÃO DAS PARTES ROBÓTICAS E PROGRAMAÇÃO INTEGRAL	33
REFERENCIAL.....	37

1 - APRESENTAÇÃO

Prezado(a) professor(a), este material didático foi desenvolvido com o intuito de auxiliá-lo(a) durante as aulas de geometria espacial, especificamente, quanto ao ensino de Prismas Retos, pelo qual, utiliza-se a Robótica Educacional, com a construção de protótipos robotizados, como recurso, para tornar as aulas de geometria mais atrativas aos estudantes, motivando-os a estudar o conceito de Prismas Retos na construção de um objeto de seu interesse. Trata-se de um produto educacional, gerado a partir de minha dissertação de mestrado, intitulada: “Robótica Educacional – Possibilidades para o ensino de Prismas Retos”, que utiliza a robótica educacional, por meio de uma sequência de atividades, durante a construção e a programação de um protótipo robô direcional, como recurso ao ensino deste conteúdo, referente a planificação, o reconhecimento e o estudo de seus elementos, a partir da confecção das partes robóticas (cabeça, tronco e membros), a ser aplicado durante sua abordagem em Geometria Espacial, do 1º ao 3º ano do ensino médio, EM13MAT309, (BRASIL 2018, p. 536), com possibilidades ao trabalho interdisciplinar entre matemática e ciências (física – eletricidade), procurando buscar, ainda, o incentivo ao pensamento computacional, pelos quais serão utilizados o software Scratch for Arduino – S4A, com a programação em blocos, (BRASIL, 2018, p.474), juntamente com a placa controladora, Arduino-Uno, e seus componentes, em ambientes sem laboratório de informática, onde, com o mínimo de um laptop, o professor poderá alçar a programação por equipes de engenharia, conforme a pesquisa da dissertação citada, pela qual conseguiu-se detectar e sanar problemas recorrentes ao ensino e aprendizagem destes tópicos da geometria, bem como incentivar e motivar os estudantes quanto a importância dos elementos, das formas e dos cálculos geométricos no dia-a-dia.

Desta forma, procuraremos, de forma sucinta, apresentar o software Scratch for Arduino – S4A, sua instalação e sincronismo com a placa arduino-uno, para então desenvolver este produto.

2 - SCRATCH, ARDUINO E SCRATCH FOR ARDUINO – S4A.

O Scratch foi concebido no Lifelong Kindergarten Group do Massachusetts Institute of Technology (MIT). É uma linguagem de programação gratuita, em um ambiente gráfico inspirada no Logo e Squeak (Etoys), que possibilita ao professor criar aulas experimentais como forma de desenvolver o raciocínio lógico e a capacidade crítica do aluno. Além da possibilidade que o aluno tem em criar experiências, compartilhar, testar e analisar seus resultados em um ambiente virtual.

O Arduino nasceu na Itália, no Interaction Design Institute Ivrea (IDII), em uma pós-graduação de design de interação focada em como as pessoas interagem com produtos, sistemas e ambientes digitais, e como eles, por sua vez, nos influenciam. A plataforma Arduino possui dois componentes principais: a placa Arduino e o Integrated Development Environment – IDE¹ (software Arduino). O primeiro é uma pequena placa micro controladora, ou seja, um pequeno circuito (a placa) que contém um computador inteiro dentro de um pequenino chip (o microcontrolador); o segundo, IDE do Arduino, é um software de desenvolvimento integrado, o qual é apresentado como uma interface gráfica de usuário (GUI), utilizado para criar um programa de computador (sketch²), em linguagem de programação C, ou ainda carregar um sketch já existente, o qual dará ordens a placa a partir do momento em que o upload do sketch para a placa for realizado.

Há muitas versões para a placa, sendo que a versão utilizada nesta pesquisa é o Arduino Uno, R3, figura 01, que é uma versão simples, resistente e de baixo custo, adequada para iniciantes em criação de protótipos.

Figura 01 - Placa Arduino Uno – R3.



Fonte: Foto da placa.

¹ IDE é um software para criar aplicações.

² Sketch – programa de computador desenvolvido pelo usuário.

A placa Arduino UNO, R3, possui 14 pinos digitais de entrada/saída (dependendo do que for especificado pelo sketch), que são enumerados de 0 à 13, dos quais: 6 pinos (3, 5, 6, 9, 10 e 11) podem ser reprogramados para saída analógica, através do programa criado - sketch; e 6 pinos de entrada analógica, enumerados de 0 à 5 (A0, A1,...,A5). Os pinos digitais recebem valores referentes aos estados de ligado ou desligado, o que, em termos elétricos, se convertem em valores de 0 ou 5V, sem nenhum valor no intervalo entre eles. Já os analógicos, recebem valores referente a tensão de um sensor, convertendo-os em um número entre 0 e 1023, os quais representam voltagens entre 0 e 5 volts, ou seja, se a tensão for 2,5V, o número que será retornado na entrada analógica será 512. Cada pino de saída do Arduino funciona com 5 volts (V). A placa arduino poderá vir, dependendo do fabricante, com uma ou duas saídas diretas de 5V, uma saída direta de 3,3V duas ou três saídas GNDs (Terra da placa, tomado como referência ao ponto neutro).

De uma forma geral:

- A entrada digital permite a leitura do estado em que se encontram os sensores, por exemplo, ligado ou desligado.
- A entrada analógica permite a leitura de sensores que enviam um sinal contínuo, como por exemplo, um sensor de som;
- A saída digital permite controlar um sensor, como por exemplo, controlá-lo em relação à quando ligar e quando desligar, permite ainda controlar motores e emitir sons, entre outros.
- A saída analógica permite-nos controlar algo contínuo, como por exemplo, a velocidade de um motor.

O S4A possui uma linguagem de programação do tipo drag and drop (arraste e solte). Este tipo de linguagem disponibiliza blocos para construir os comandos e estes são criados através de ações com o mouse. Permite o trabalho com seis entradas analógicas, duas entradas digitais, três saídas analógicas, três saídas digitais e quatro saídas para conectar servomotores de rotação, ou seja, ao conectar os condutores na placa Arduino, precisamos estar atentos aos blocos e pinos disponíveis no S4A para serem programados.

Para utilizar-se a placa Arduino no S4A, faz-se necessário o upload de um sketch / firmware, uma única vez, referente a essa comunicação entre eles, cuja

programação esteja direcionada, para que a linguagem utilizada pelo S4A seja traduzida para a linguagem compreendida pela placa Arduino, sempre que a programação criada for executada. No site S4A.cat, figura 2, é possível fazer o download do software e do firmware³ utilizado para o funcionamento e o sincronismo entre a placa e o S4A.

Figura 2 - Site S4A.cat



Fonte: <http://s4a.cat/>

Antes da instalação/execução do S4A, deve-se carregar a placa Arduino com o firmware baixado. Para isso, com a placa conectada ao computador, deve-se abrir o IDE, depois ir ao menu arquivo e em seguida abrir. Procurar a pasta onde o firmware foi armazenado e abri-lo no GUI, conforme figura 3.

Figura 3 - Firmware Arduino.

```
© S4AFirmware16 | Arduino 1.6.15
Arquivo: Editor Sketch Ferramentas Ajuda
S4AFirmware16
// New in version 1.6.0 (by Jorge Gonçalves)
// Fixes variable type in pin structures pin.state should be int, not byte
// Optimized speed of execution while removing data from computer in readSerialLine()

// NEW IN VERSION 1.6.0 (by Jorge Gonçalves)
// Added new structure actuator to hold the pins information
// - This makes the code easier to read and modify (MDN)
// - Allows to change the type of pin more easily to meet non standard use of S4A
// - Eliminates the need of having to deal with different kind of index access (ie: states[pin-4])
// - By using an array we hold all the possible output pin states the code is now more readable
// Changed all functions using old style pin access: configurePin(), resetPin(), readDigitalPort(), updateActuator() and sendUpdateActuator()
// Fixed possible overflow every 70 minutes (2^32 us) in pulse() while using micros(), changed for delayMicroseconds()
// Some minor coding style fixes

// NEW IN VERSION 1.6.0 (by Jorge Gonçalves)
// Fixed compatibility with Arduino Leonardo by avoiding the use of timers
// readSerialLine() optimized
// - Created state machine for reading the two bytes of the S4A message
// - updateActuator() is only called if the state is changed
// Minify use optimization
// Cleaning some parts of code
// Avoid using some global variables

// NEW IN VERSION 1.6.0
// Refactored reset pins
// Merged code for standard and IO matrix
// Special thanks for feedbacks from Peter Mueller (many thanks for this!!)

// NEW IN VERSION 1.6.0
// changed pin A from standard serial to normal digital output

// NEW IN VERSION 1.6.0
```

Fonte: IDE do software.

³ Firmware é um software para o controle de dispositivos eletrônicos.

Com a placa conectada, clique no ícone carregar, figura 4, o que possibilitará o armazenamento das informações para o funcionamento da placa Arduino com o S4A dentro dos limites do firmware utilizado, ou seja, de acordo com o projeto a ser desenvolvido, poderá haver necessidade de firmwares mais específicos, os quais poderão ser buscados nas diversas bibliotecas online referentes ao S4A.

Figura 4 - Ícone carregar.



```

S4AFirmware16 | Arduino 1.8.13
Arquivo Editar Sketch Ferramentas Ajuda
Carregar
S4AFirmware16
// Merged patch for Leonardo from Peter Mueller (many thanks for this!)

// NEW IN VERSION 1.5:
// Changed pin 8 from standard servo to normal digital output

```

Fonte: IDE do software.

Após o carregamento, o IDE deverá ser fechado, para evitar conflitos, e em seguida continuar com a instalação e execução do S4A.

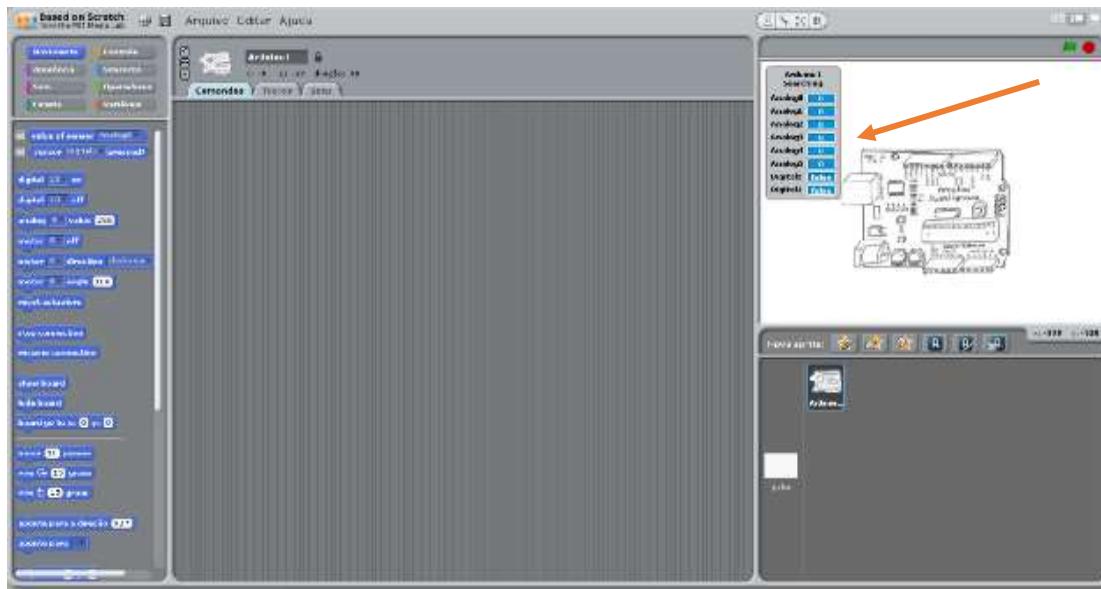
Com a software aberto e a placa conectada ao computador, ao lado direito da interface deverá aparecer uma janela com as portas oscilando valores numéricos, conforme figura 5, o que indica que a placa e o software estão se comunicando, caso contrário, estará sem comunicação, figura 6.

Figura 5 - Placa com comunicação.



Fonte: Interface do software

Figura 6 - Placa sem comunicação.



Fonte: Interface do software

Componentes:

Para o controle do robô direcional serão utilizados: 01 placa Arduino Uno R3; 03 Micro Servomotor 9G SG90; 02 motores DC 3 – 6V com redução; 01 Chassi Carro para Arduino; 02 leds auto brilho vermelhos (3v) e 04 leds auto brilho azuis (3v); 01 protoboard pequena; 01 Módulo Ponte H dupla HG 7881; 8 jumpers médios “macho-fêmea”; 11 jumpers médios “macho-macho”, 01 bateria de 9V com clip conector “plug rabicho” e 04 pilhas alcalinas de 1,5v cada com suporte.

Nota: Em caso de utilização de sucatas, os estudantes poderão utilizar no lugar dos motores DCs, motores de DVDs ou aparelhos semelhantes, com observação às engrenagens para o movimento do motor, caso o robô seja pesado. Da mesma forma, a base direcional poderá ser confeccionada com restos de materiais, observando o limite para as pernas robóticas e placa controladora.

3 – A ROBÓTICA EDUCACIONAL

A Robótica Educacional surgiu com a criação da linguagem de programação LOGO, desenvolvida por Seymour Papert em meados de 70, a qual era utilizada em um computador conectado a um robô, denominado Tartaruga de Chão (Floor Turtle),

figura 52, por meio de um longo “cordão umbilical”, em que, a partir das instruções escritas em LOGO, a mesma traçava figuras simples no chão (RESNICK, 1996).

A Robótica Educacional é o desdobramento natural das crenças e dos trabalhos de seu criador, Seymour Papert, que em meados de 70 desenvolveu um protótipo robô, denominado Tartaruga de chão (Floor Turtle) ligado a um computador, por meio de um longo fio, que através das instruções programadas em LOGO, desenvolvia atividades voltadas à aulas de matemática, geometria.

A aula geralmente é direcionada para a construção de um protótipo e, posteriormente, é feita a programação deste, considerando sempre a idade e a capacidade cognitiva de cada aluno. As atividades de robótica educacional motivam e encorajam os alunos a resolverem problemas autênticos que são significativos para eles, proporcionando-lhes a oportunidade de vivenciarem a experiência de buscar e encontrar soluções.

Segundo Papert (1986) na Robótica Educacional:

- O projeto deve ter por base a construção de objetos físicos.
- Os objetos construídos pelos alunos devem ser programados por eles.
- Em cada projeto o aluno deve criar seus objetos de estudos de maneira livre e colaborativa.

A robótica educacional é, portanto, um ambiente de ensino e aprendizagem baseado na construção de um protótipo, robô, com componentes eletromecânicos e processadores ligados a um software de computador, cujas características, definidas por Seymour Papert, são projetos em que os alunos possam criar, construir e programar seus próprios objetos de estudo.

4 – PROBLEMA PROPOSTO

Antes de repassar as atividades aos estudantes você deverá direcioná-lo a um robô específico, preferencialmente do próprio conhecimento deles, procurando instiga-los sobre seu formato, deixando-os à vontade. Nesse caso, o nosso robô será baseado nos personagens do jogo Minecraft, bastante conhecido entre os jovens.

Deverão ser formadas 04 equipes, e o seguinte direcionamento deverá ser repassado:

Um dos jogos mais conhecidos no mundo é o Minecraft, onde os jogadores exploram um mundo aberto tridimensionalmente, cujos objetos e personagens são basicamente formados por blocos de faces retangulares, conforme figura 54.

Figura 54 - Print do jogo Minecraft.



Fonte: Wikepedia.

Para cada grupo de estudantes serão entregues responsabilidades e atividades específicas direcionadas ao protótipo proposto:

- Grupo 01 – Responsável pelos braços robóticos;
- Grupo 02 – Responsável pelas pernas robóticas;
- Grupo 03 – Responsável pelo tronco robótico;
- Grupo 04 – Responsável pela cabeça robótica;

Desta forma, os estudantes deverão ser direcionados à mesma regra da construção dos blocos retangulares, e a partir das responsabilidades de cada grupo, desenvolver as seguintes atividades:

4.1 - ATIVIDADE I – Construir um protótipo robô em formato de blocos retangulares, com as sugestões: suas faces (cabeça, tronco e membros) devem, além do formato em questão, obedecerem a características específicas.

Nesta atividade o professor deverá frisar aos estudantes a importância das medidas corretas para o formato sugerido, além de atentarem-se às faces que ficaram abertas para a acoplagem dos servomotores.

Equipe Braços:

- ✓ Os dois braços devem ter formatos e comprimentos iguais e serem diferentes do formato das pernas, com medidas da base 5cm de largura por 8cm de comprimento e altura 12cm.

Equipe Pernas:

- ✓ As duas pernas devem ter formatos e comprimentos iguais, com medidas da base 5cm por 5cm e altura 7cm.

Equipe Tronco:

- ✓ O tronco deverá ter a base medindo 10cm de largura por 12cm de comprimento e altura 16cm.

Equipe Cabeça:

- ✓ A cabeça deverá ter todas as arestas medindo 8cm.

4.2 - ATIVIDADE II – Em cada protótipo construído, quando necessário, acoplar: motor DC, servomotor, leds e/ou motores de sucata, para acionamento e realização dos movimentos, descrevendo o algoritmo para sua programação.

Nessa atividade o professor deverá alertar os estudantes à posição das palhetas dos servomotores, assim como as posições das fiações dos motores DCs, para que não haja movimentos aleatórios.

Equipe Braços:

- ✓ Acoplar um servomotor em cada braço, ajustando suas posições para movimentos horário e anti-horário.

Equipe Pernas:

- ✓ Acoplar dois motores DC's à base direcional, com as respectivas ligações dos jumpers entre Ponte H e placa Arduino.

Equipe Tronco:

- ✓ Montar um circuito em paralelo, com 4 leds, e acoplagem dos mesmos em cada vértice frontal do tronco, com as respectivas ligações na placa arduino.

Equipe Cabeça:

- ✓ Na face frontal da cabeça, acoplar 02 leds, de mesma cor, ligados no circuito em paralelo, na posição dos olhos robóticos, e um servomotor na posição do pescoço;

4.3 - ATIVIDADE III – Realizar a programação utilizando o Scratch for Arduino – S4A, para o funcionamento de cada protótipo construído.

Nesta atividade o professor levará o laptop até a equipe que for terminando a construção de seu protótipo, com os respectivos motores e/ou circuitos acoplados, para realizarem a programação, a qual será debatida por todos e manuseada por apenas um componente.

Equipe Braços:

- ✓ Desenvolver a programação para Movimento dos Braços (horário e anti-horário).

Equipe Pernas:

- ✓ Desenvolver a programação para o movimento direcional das Pernas (frente, trás, direita e esquerda).

Equipe Tronco:

- ✓ Desenvolver a programação para acionamento dos leds do tronco robótico.

Equipe Cabeça:

- ✓ Desenvolver a programação para o movimento da cabeça (intervalos de 0º a 30º).

4.4 - ATIVIDADE IV – Acoplar e reunir as sintaxes de programação, com um componente de cada grupo, para o funcionamento integral do protótipo Robô, da seguinte forma:

- ✓ Reunir as sintaxes de programação da cabeça e dos braços junto a sintaxe do movimento direcional (pernas);
- ✓ Juntar os circuitos dos olhos e dos vértices frontais e ligar direto à porta de 3,5V da placa arduino.

5 – CONSTRUÇÃO E PROGRAMAÇÃO DO PROTÓTIPO ROBOTIZADO.

5.1 - Cabeça Robótica.

Com o uso de uma régua, um lápis e restos de caixas de papelão, inicia-se a construção do formato sugerido, cubo, a partir de sua planificação, estabelecendo as medidas das arestas e delimitando suas faces, considerando a proporção com as demais partes do corpo robótico. A face superior do sólido construído deverá ficar aberta, figura 7, para instalação e manuseio do circuito elétrico (olhos robóticos) e servomotor (pescoço robótico).

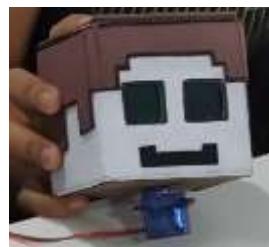
Figura 7 – Construção da cabeça em formato de cubo.



Fonte: Arquivo pessoal.

Uma vez realizada a construção da cabeça em formato de cubo, deverá ser realizado os designers necessários às faces do robô sugerido, assim como a acoplagem do servomotor, posicionado como pescoço robótico, conforme figura 8, pelo qual suas palhetas deverão ser parafusadas e coladas, com cola quente, internamente à cabeça, face inferior do cubo, de tal forma que a mesma consiga realizar os movimentos angulares com precisão, respeitando a posição do servo.

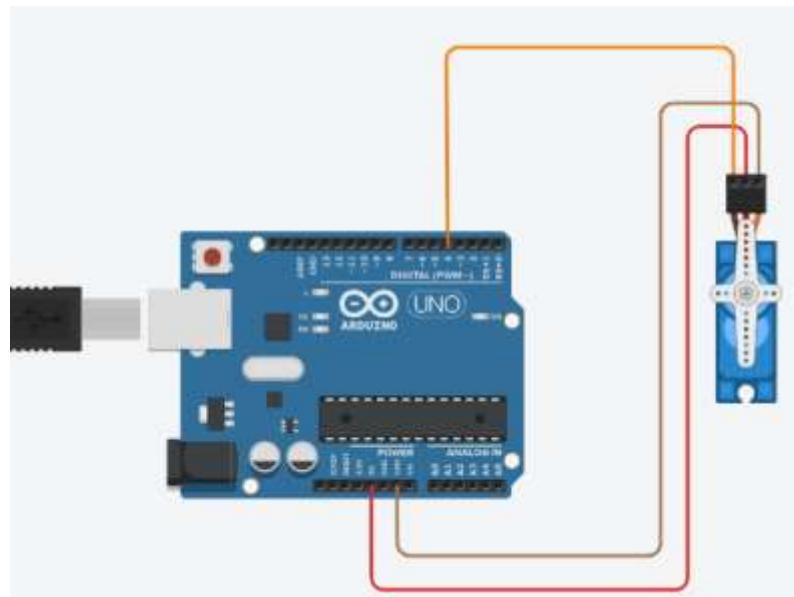
Figura 8 - Placa com comunicação.



Fonte: Arquivo pessoal.

Com o servomotor acoplado à cabeça robótica, realiza-se as ligações dos jumpers (macho-macho) do servo à placa arduino, com o vermelho ligado à saída de 5v, o marrom ligado ao GND, e o laranja ligado à porta serial 4, conforme figura 9.

Figura 9 – Ligação do servomotor à placa controladora.

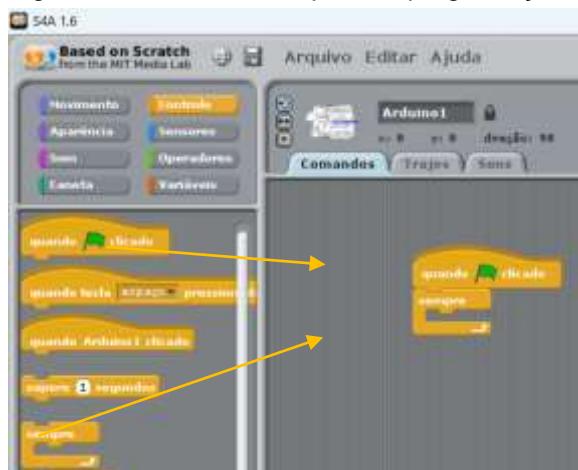


Fonte: Simulação realizada no software Tinkercad.

Com a placa conectada, a cabeça estará pronta para receber a programação. De posse do computador e com o software S4A aberto, realiza-se a ligação da placa ao computador, por meio da porta USB, e, com o reconhecimento da placa pelo software, inicia-se a programação.

O S4A, apresenta 08 ferramentas de blocos programáveis, separados por cores e com ações específicas ao que se pretende desenvolver. Pelos quais, para que se dê início a uma programação, utiliza-se a ferramenta “Controle”, por meio da qual, utiliza-se o bloco “quando bandeira clicado”, o arrastando para a interface de programação, juntamente com um loop, bloco “sempre”, para que a placa desenvolva sempre a sintaxe desenvolvida, conforme figura 9.

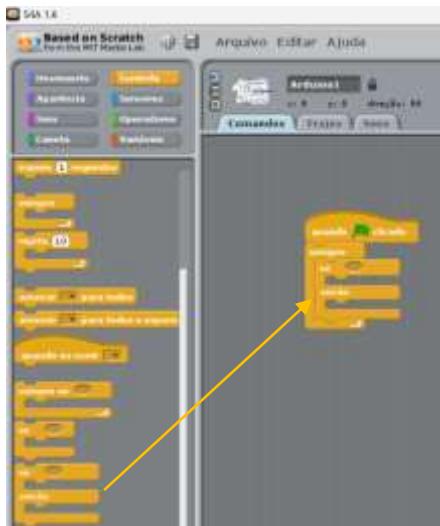
Figura 9 – Blocos iniciais para as programações.



Fonte: Print do software S4A.

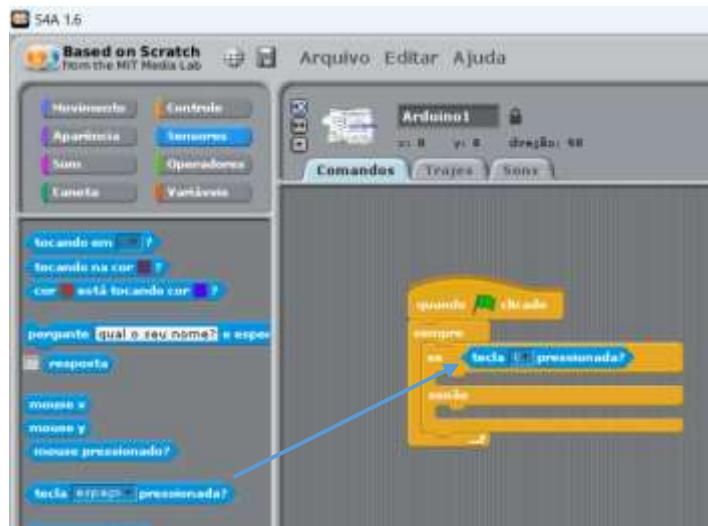
Após os blocos de partida posicionados, precisa-se mandar a informação sobre qual tecla deverá acionar o movimento da cabeça. Para isso, ainda em “controle”, utiliza-se o bloco condicional “Se senão”, figura 10, e em seguida utiliza-se a ferramenta “Sensores”, arrastando o bloco “tecla espaço pressionada” para a posição da primeira condicional, o qual deverá ser alterado, clicando na seta vertical do bloco, a tecla “espaço” pela tecla “c”, conforme figura 11.

Figura 10 – Bloco condicional.



Fonte: Print do software S4A.

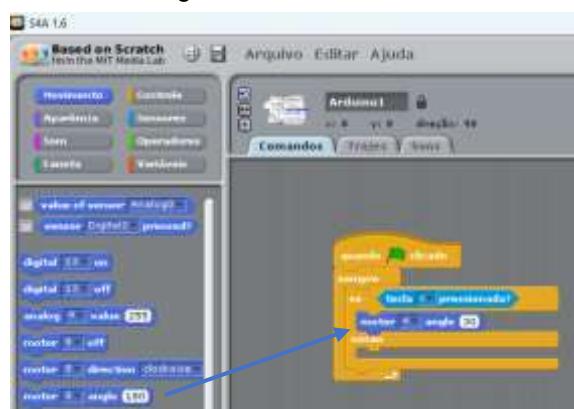
Figura 11 – Ferramenta Sensores.



Fonte: Print do software S4A.

Uma vez informada a tecla a ser pressionada, acrescenta-se o bloco referente ao servomotor com o respectivo movimento angular. Para isso, utiliza-se a ferramenta “Movimento” e arrasta-se o bloco “motor 8 angle 180°”, pelo qual deve-se alterar, utilizando a seta, a porta 8 para porta 4 e o ângulo de 180° para 30°, conforme figura 12.

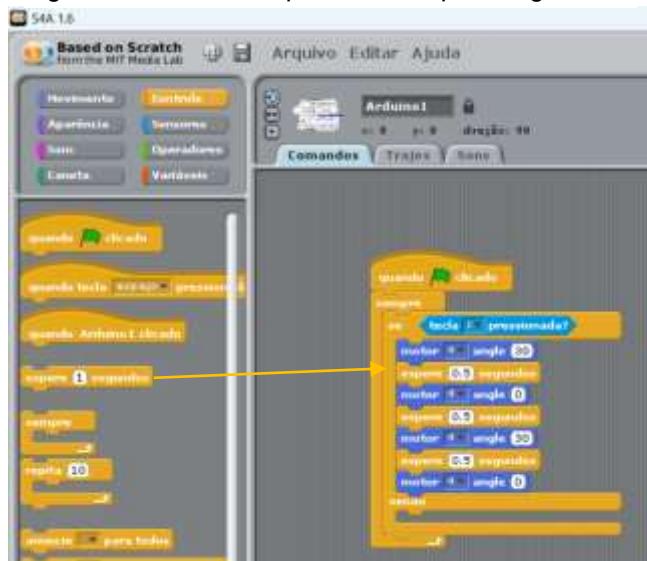
Figura 12 – utilização do bloco motor para o movimento angular.



Fonte: Print do software S4A.

Para que ocorra o movimento angular de ida e volta, deve-se acrescentar um bloco com ângulo de 30° e outro com ângulo de 0°, com um bloco temporizador entre eles. Para isso, na ferramenta “Controle”, utilizar o bloco “espere 1 segundo”, o qual deverá ser alterado para “0,5 segundo” e replicá-lo pelo menos duas vezes, conforme figura 13.

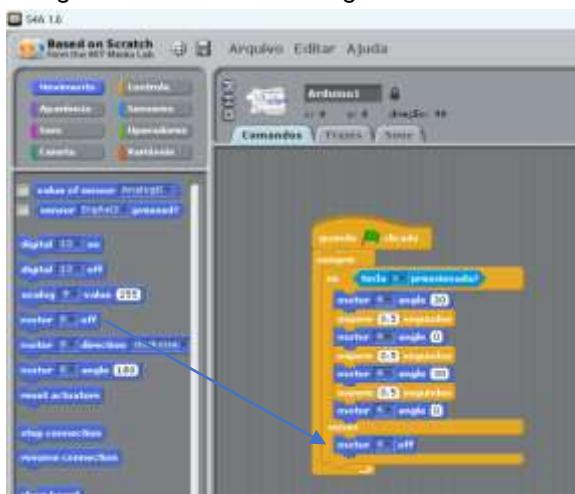
Figura 13 – Bloco temporizador “espere segundos”.



Fonte: Print do software S4A.

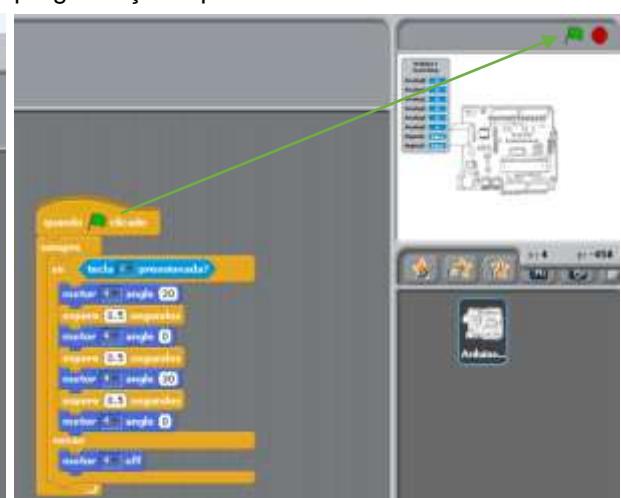
Para concluir a programação, adiciona-se na segunda condicional, o bloco de desligamento, “motor 4 of”, figura 14, passando, por conseguinte, à compilação, clicando na bandeirinha, conforme figura 15, responsável por carregar a sintaxe de programação à placa controladora, a qual, a partir da tecla “c” pressionada realizará o movimento da cabeça robótica de 0° a 30° .

Figura 14 – Bloco de desligamento do motor.



Fonte: Print do software S4A.

Figura 15 – Bandeirinha para carregamento da programação à placa controladora.



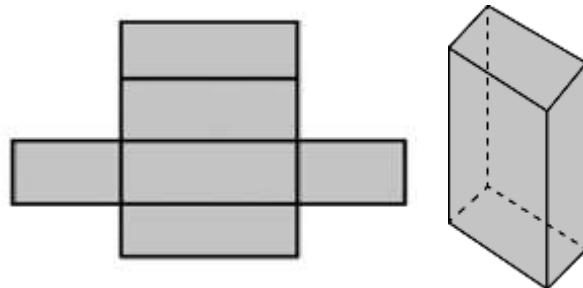
Fonte: Print do software S4A.

5.2 - Braços Robóticos.

Os braços robóticos terão o formato de paralelepípedo retângulo, com medidas proporcionais ao tronco e a cabeça robótica, utilizando uma régua centimétrica e

construídos a partir de sua planificação, figura 16. A face superior e uma das faces laterais deverão estar abertas para a instalação dos servomotores, pelos quais, as palhetas deverão ser posicionadas, parafusadas e coladas com cola quente, internamente aos braços, observadas a posição junto ao tronco, o qual comportará a parte externa do servomotor até o limitador, conforme figura 17.

Figura 16 – Planificação e montagem do braço robótico.



Fonte: Autor.

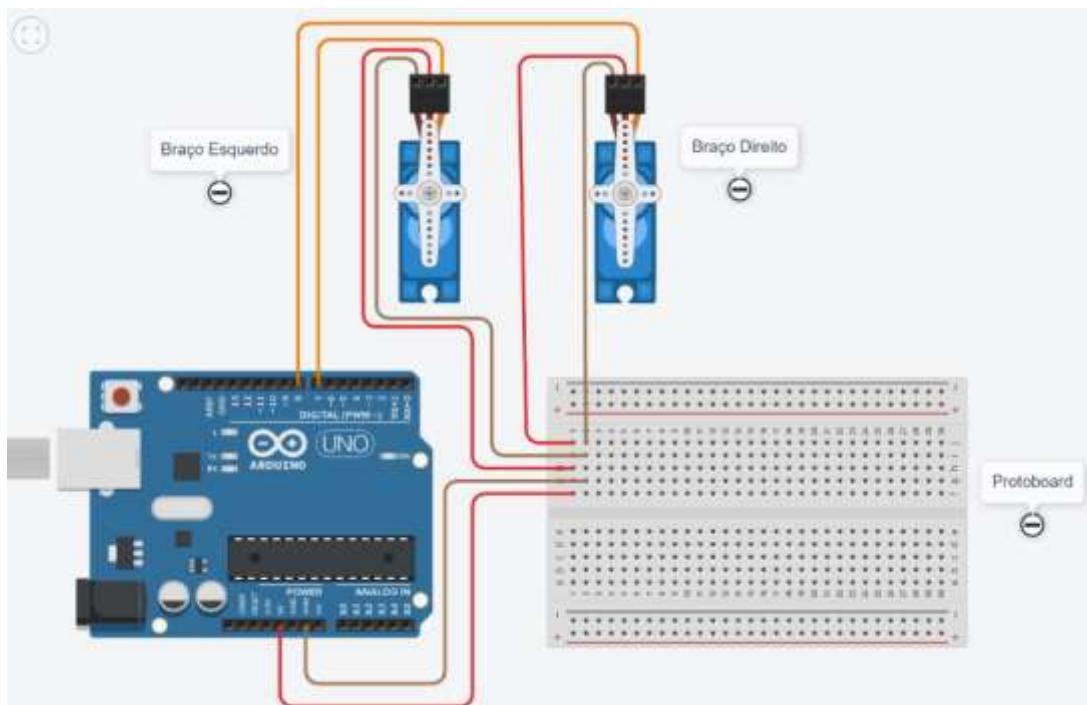
Figura 17 – Acoplagem dos braços robóticos ao tronco robótico.



Fonte: Arquivo pessoal.

Com os braços robóticos construídos, passa-se a ligação dos jumpers, referentes aos dois servos, à placa controladora, pelos quais será utilizado uma protoboard para a transferência das saídas de 5v e GND, conforme figura 18. A conexão à placa arduino é semelhante a realizada com a cabeça robótica, alterando apenas as portas seriais, que ficarão na 7 e na 8.

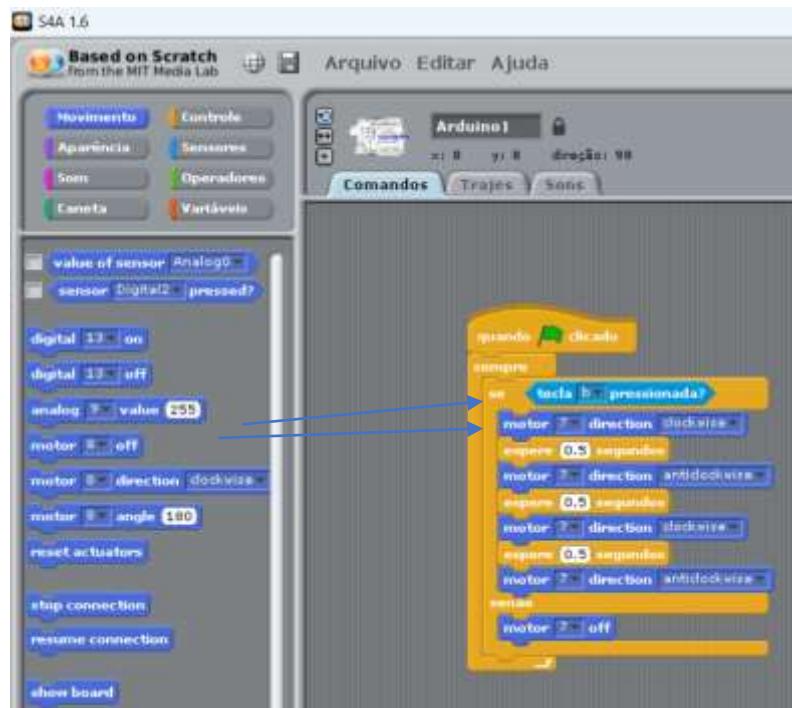
Figura 18 – Ligação dos jumpers dos servomotores dos braços, com fonte de alimentação transferida à protoboard.



Fonte: Simulação realizada no software Tinkercad.

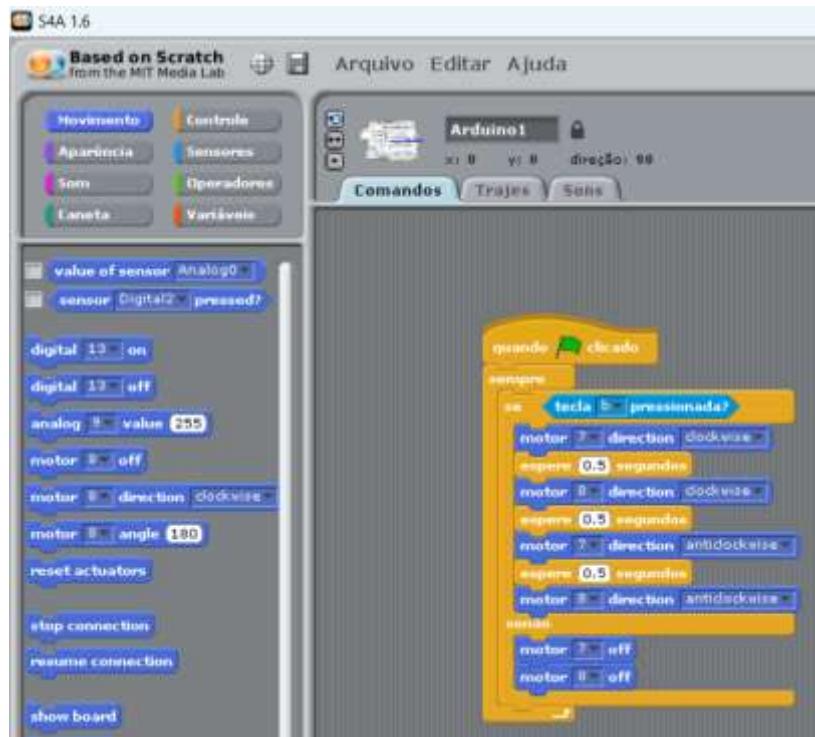
Com os jumpers conectados, segue-se para a programação, com as mesmas prerrogativas da cabeça robótica, ou seja, realiza-se a programação para um braço, semelhante a realizada para a cabeça, alterando apenas a porta serial e os ângulos de rotação, que agora serão horário e anti-horário a partir da tecla “b” pressionada, figura 19, e depois acrescenta-se a sintaxe do segundo braço, com sua respectiva porta, cuja programação final seja o acionamento dos dois braços robóticos a partir da tecla “b” pressionada, figura 20.

Figura 19 – Movimentos horário e anti-horário.



Fonte: Print do software S4A.

Figura 20 – Sintaxe para o movimento dos dois braços.

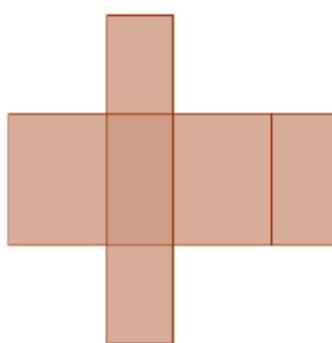


Fonte: Print do software S4A.

5.3 - Tronco Robótico.

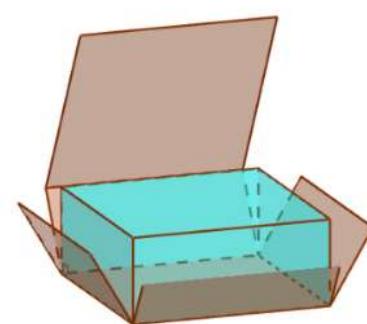
O tronco robótico terá o formato de paralelepípedo retângulo, com medidas proporcionais a cabeça robótica. Em sua construção será utilizado restos de papelão, uma régua centimétrica, fios eletrônicos (0,5mm) e 04 leds, onde, semelhante às outras construções, seguirá os moldes da planificação, observando as medidas das arestas e faces, figura 21 e 22, os revestimentos do design sugerido, figura 23.

Figura 21 – Construção do sólido planificado.



Fonte: Autor

Figura 22 – Simulação de como ficará o sólido construído.



Fonte: Autor

Figura 23 – Tronco robótico revestido e com os leds.



Fonte: Arquivo pessoal.

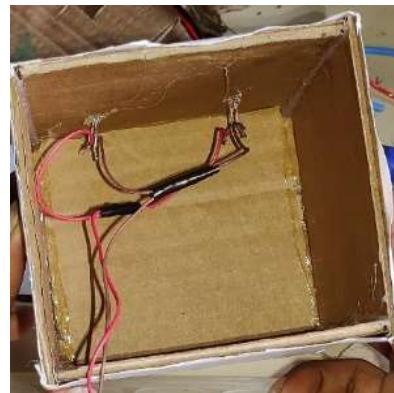
As faces, superior e inferior, deverão ficar abertas para o manuseio dos leds do circuito elétrico, azuis e vermelhos, os quais deverão ser posicionados nos vértices frontais do sólido construído, figura 23; para a acoplagem dos servomotores referentes aos braços e cabeça robótica, assim como, para a fixação das hastes que prendem o tronco às pernas robóticas.

5.3.1 - Circuito Elétrico.

Para o circuito elétrico será utilizado dois fios eletrônicos (0,5mm) de cores preta e vermelha, de 60cm cada, onde será utilizada a cor preta para o polo negativo e vermelha para o positivo, juntos a 06 leds de 3v cada (04 azuis e 02 vermelhos).

Serão construídos dois circuitos, um para os olhos robóticos, figura 24, e o outro para os vértices frontais do tronco, figura 25, e depois serão agrupados em único circuito, figura 26, o qual será disposto em paralelo e ligado às saídas, 3,3v e GND, da placa arduino, conforme figura 27.

Figura 24 – circuito olhos robóticos.



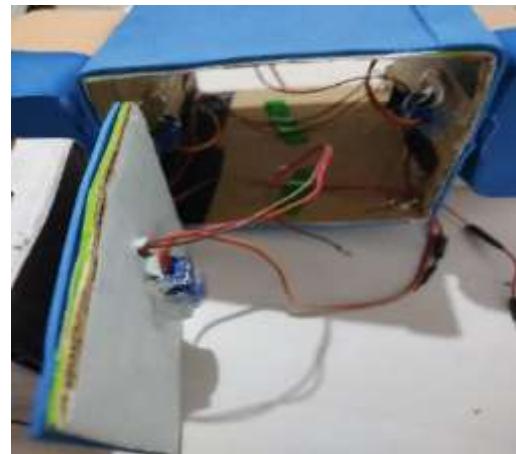
Fonte: Arquivo pessoal.

Figura 25 – circuito dos vértices frontais.



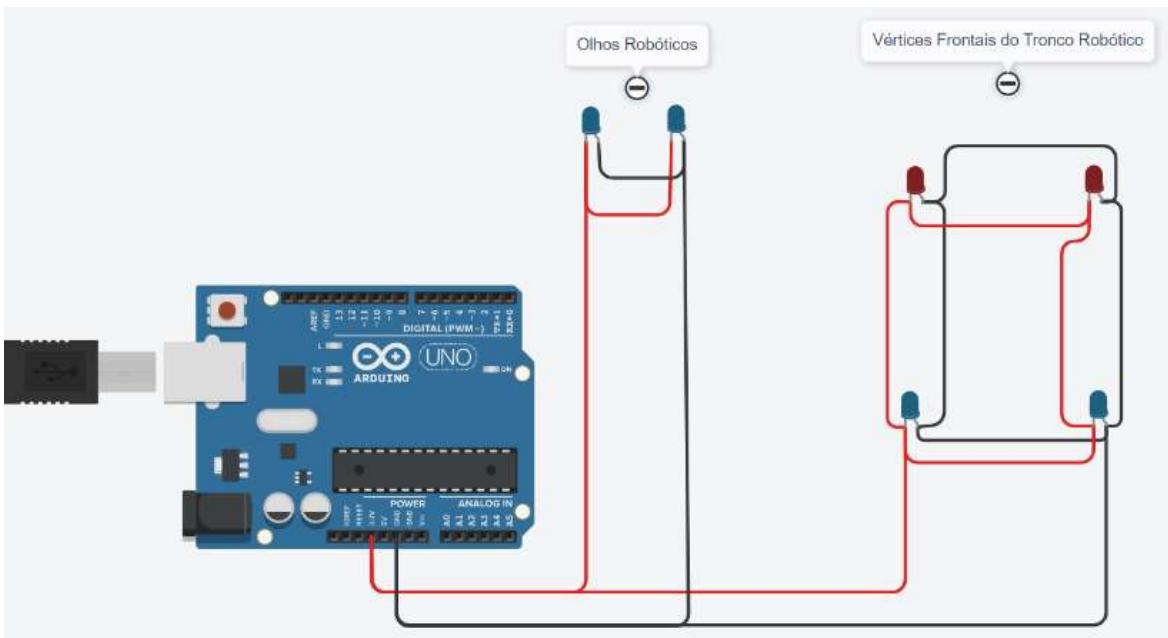
Fonte: Arquivo pessoal.

Figura 27 – Junção dos circuitos: olhos e vértices.



Fonte: Arquivo pessoal.

Figura 27 – Ligação dos circuitos, olhos e vértices, à placa arduino.



Fonte: Simulação realizada no software Tinkercad.

Para o caso em que se queira programar o acionamento dos leds, ou ainda agregar mais conhecimentos da física, haverá a necessidade de se usar um resistor, uma vez que as saídas das portas seriais fornecem 5V e os leds são de 3V. Para isto, por meio da lei de Ohm deve-se calcular o resistor necessário, calculando sua resistência e potência, considerando os leds de alto brilho de 20mA.

Como os leds estarão ligados num circuito em paralelo, bastará calcular o resistor considerando um led, que servirá para os demais.

Como a placa fornece a saída de 5V e os leds suportam 3V, obtém-se a sobra para o cálculo do resistor:

$$V_{resistor} = V_{placa} - V_{led}$$

$$V_{resistor} = 5 - 3$$

$$V_{resistor} = 2\text{V}$$

Portanto:

$$R = \frac{V}{I} \quad R = \frac{2}{0,02} \quad R = 100\Omega$$

$$P = \frac{V^2}{R} \quad P = \frac{2^2}{100} = P = 0,04W = 40mW$$

Com R = resistência.

P = Potência.

I = Corrente elétrica.

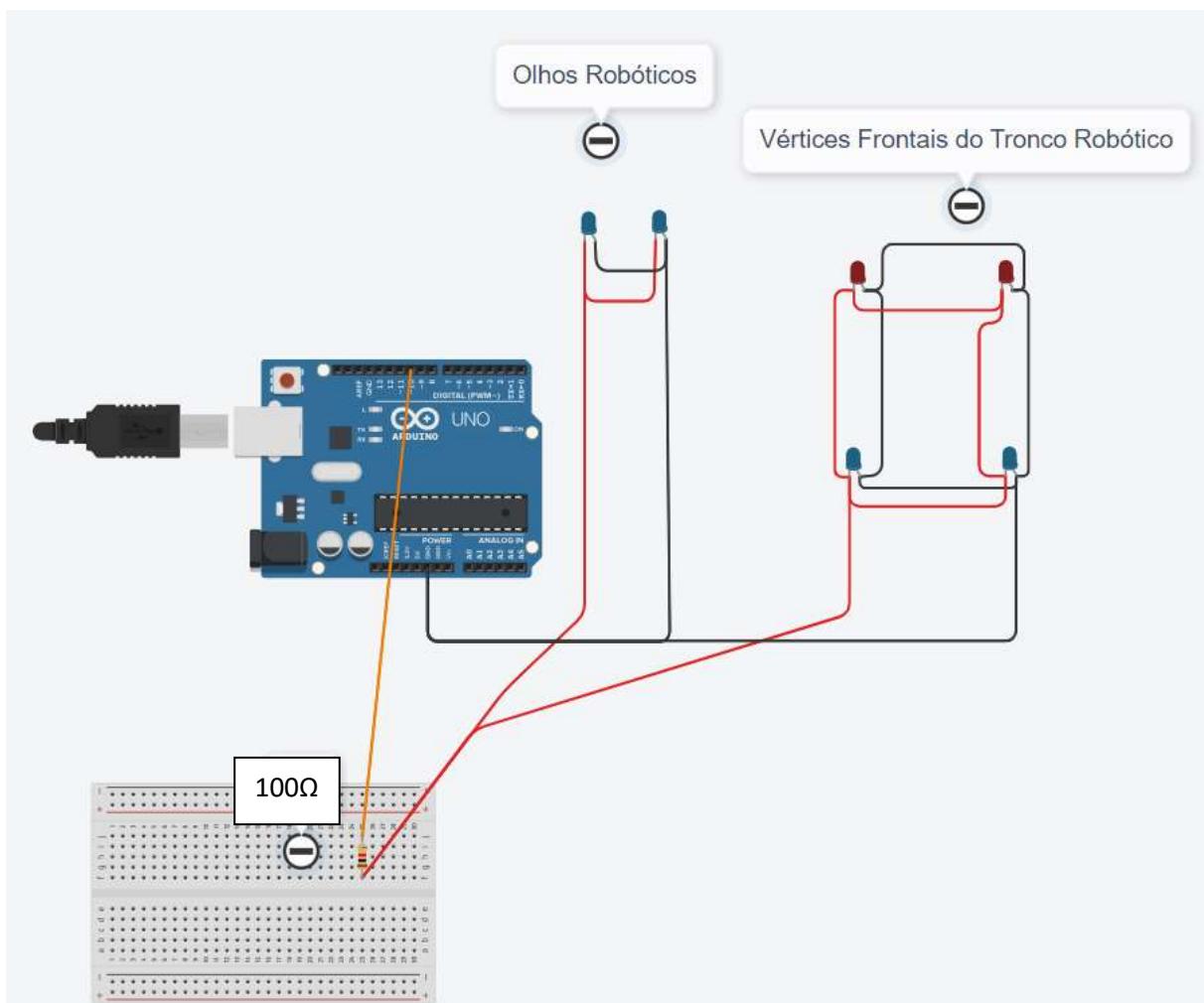
V = tensão.

Como são 6 leds em paralelo, logo a potência total será

$$P = P_1 + P_2 + P_3 + P_4 + P_5 + P_6 \quad P = 6 \times 40 \quad P = 240mW$$

Logo, o circuito estará ligado a um resistor de no mínimo 100Ω , potência $1/4W$, com polo positivo ligado à saída serial da porta 10 e negativo no GND, conforme figura 28.

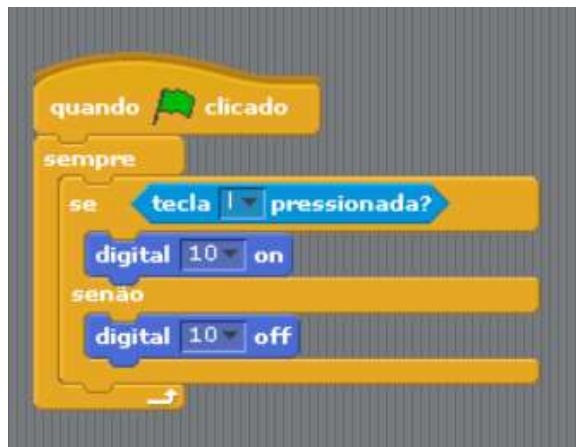
Figura 28 – Utilização do resistor para a programação dos leds



Fonte: Simulação realizada no software Tinkercad.

A programação seguirá a mesma sintaxe do servomotor, com acionamento pressionando a tecla “L”, figura 29.

Figura 29 – Programação para acionamento dos leds.



Fonte: Print do software S4A.

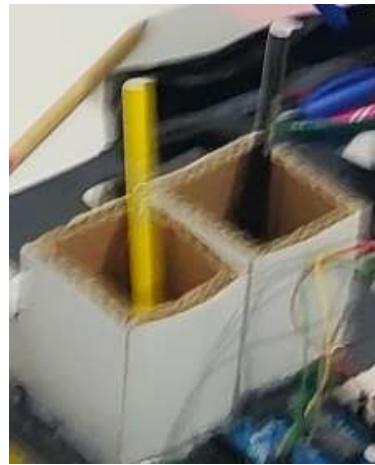
OBS.: Esta Sintaxe não será utilizada na programação final do protótipo robô.

5.4 - Pernas robóticas.

As pernas robóticas terão o formato de Prisma quadrangular regular, construídas a partir da planificação, com o uso de caixas de papelão, observando as arestas das bases e das laterais, pelas quais, terão o comprimento lateral reduzido, comparado às demais partes robóticas (cabeça, tronco e braços), haja vista, a previsibilidade dos movimentos a serem realizados pelo robô, para que o mesmo não perca o equilíbrio, assim como, as medidas das faces das bases deverão ser proporcionais à base direcional e à base do tronco.

Serão utilizados restos de caixas e uma régua centimétrica para que, a partir das planificações, sejam feitas as construções. As faces superiores das bases deverão estar abertas para que se coloque uma haste ligando, internamente, a perna ao tronco, com o intuito de proporcionar maior rigidez entre pernas e o corpo robótico, figura 30.

Figura 30 – haste de fixação entre pernas e tronco robótico.



Fonte: Arquivo pessoal.

5.4.1 - Base Direcional.

A base direcional deverá ser construída em formato retangular e medidas proporcionais ao robô construído, considerando as bases inferiores das pernas robóticas, as rodas com os respectivos motores, a roda direcional, a placa controladora, a placa Ponte H, a protoboard, a bateria de 9v e o suporte das 04 pilhas pequenas, figuras 30 e 31.

Figura 30 – Construção e acoplagem dos periféricos à base direcional.



Fonte: Arquivo pessoal.

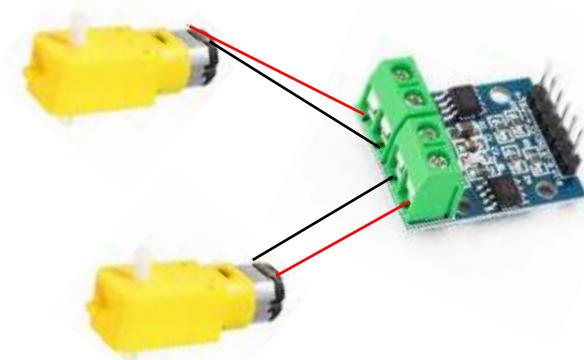
Figura 31 – fixação das pernas robóticas à base direcional.



Fonte: Arquivo pessoal.

Com a base construída, passa-se inicialmente à ligação dos motores DCs à placa Ponte H, com observação aos polos positivos de cada motor, os quais serão soldados com um fio eletrônico (0,5mm) de cor vermelha, e o negativo com o de cor preta, conforme figura 32.

Figura 32 – ligação dos motores DCs à placa ponte H.



Fonte: Autor

Com os motores ligados, deve-se tomar nota das posições, de acordo com a legenda de cada pino, quadro 1, para a posterior programação dos movimentos robóticos, a serem realizados após a conexão entre Ponte H e a placa Arduino.

Quadro 1 - Legenda dos pinos da Ponte H.

B-1A = Motor B acionado para frente.
B-1B = Motor B acionado para trás.
GND = Terra (Negativo)
VCC = Positivo
A-1A = Motor A Acionado para trás.
A-1B = Motor A acionado para frente

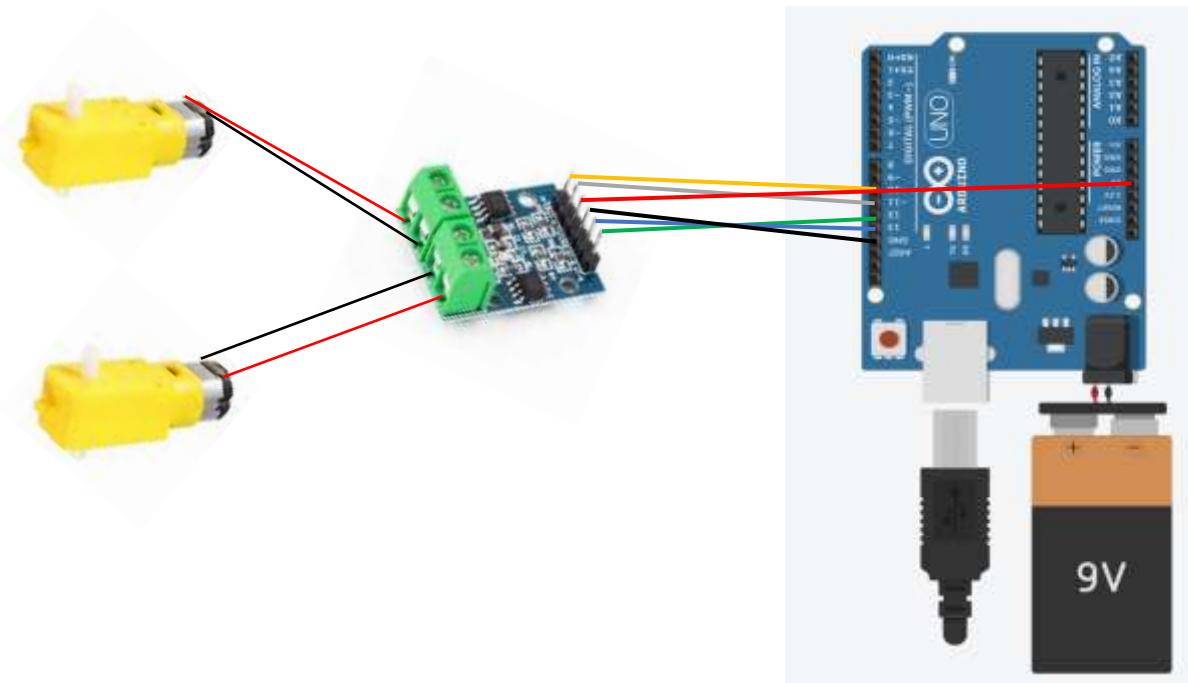
Fonte: Autor

Com as anotações feitas, posicionaremos os jumpers (fêmea-fêmea) às portas serias: 10, 11, 12 e 13, da seguinte forma:

- Movimento para Frente: Portas 10 e 12.
- Movimento para trás: Portas 11 e 13.

Portanto, as ligações deverão estar conforme figura 33.

Figura 33 – ligação das portas da placa Ponte H às portas da placa Arduino.



Fonte: Autor

Com a base direcional construída, acoplam-se as pernas robótica, a roda direcional e a dos motores DCs, conforme figuras 34 e 35.

Figura 34 – Acoplagem dos motores DCs e rodas à base direcional.



Fonte: Arquivo pessoal.

Figura 35 – Acoplagem das pernas robóticas à base direcional.



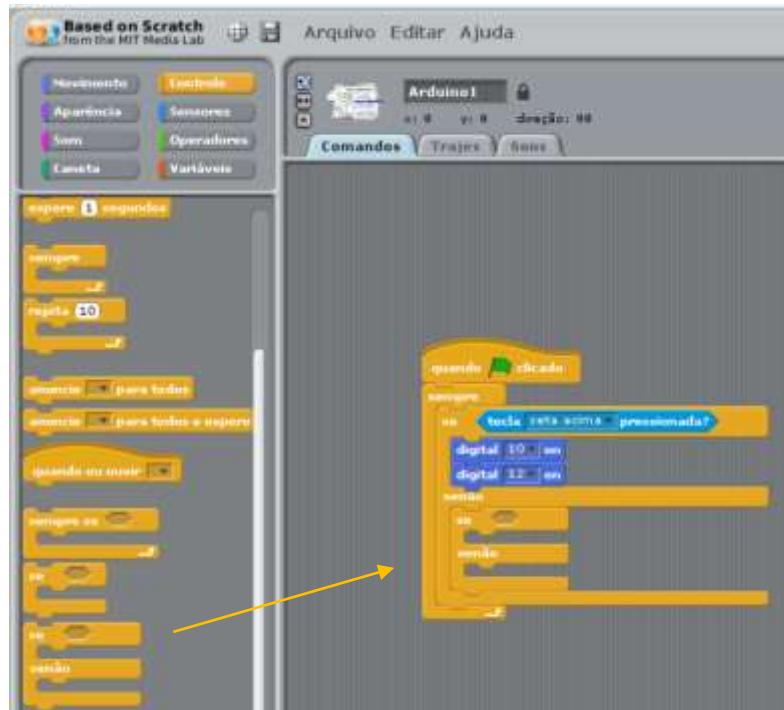
Fonte: Arquivo pessoal.

Após a montagem, segue-se com a programação, a qual deverá estar de acordo com as portas seriais ligadas a cada movimento dos motores, conforme anotação. Desta forma, os movimentos serão programados utilizando as setas do teclado da seguinte forma:

- Movimento para frente: seta pra cima – acionamento dos motores das portas 10 e 12.
- Movimento para trás: seta pra baixo – acionamento dos motores das portas 11 e 13.
- Movimento para direita: seta para direita – acionamento do motor da porta 10 e desligamento do motor da porta 12.
- Movimento para esquerda: seta para esquerda – acionamento do motor da porta 12 e desligamento do motor da porta 10.

Assim, segue-se a mesma sequência de programação dos outros protótipos, com observância às condicionais colocadas de acordo com a quantidade de movimento que se queira executar, conforme figura 36.

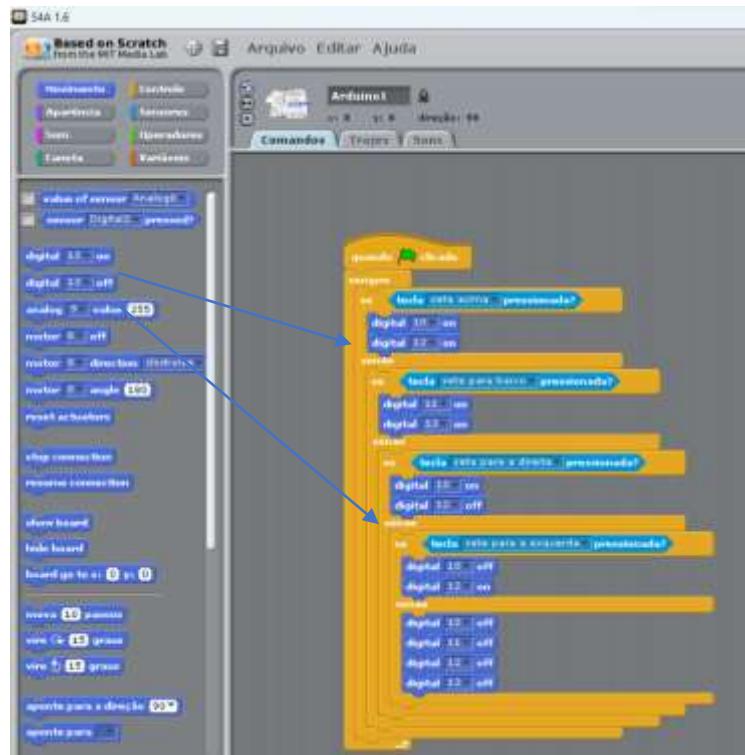
Figura 36 - Blocos condicionais para os movimentos direcionais.



Fonte: Print do software S4A.

Logo, todas as condicionais referentes aos movimentos serão acrescentadas após a segunda condicional, “senão”, figura 37, com os respectivos “Sensores”, teclas, para acionamento.

Figura 37 – Blocos condicionais e seus respectivos “Sensores”.

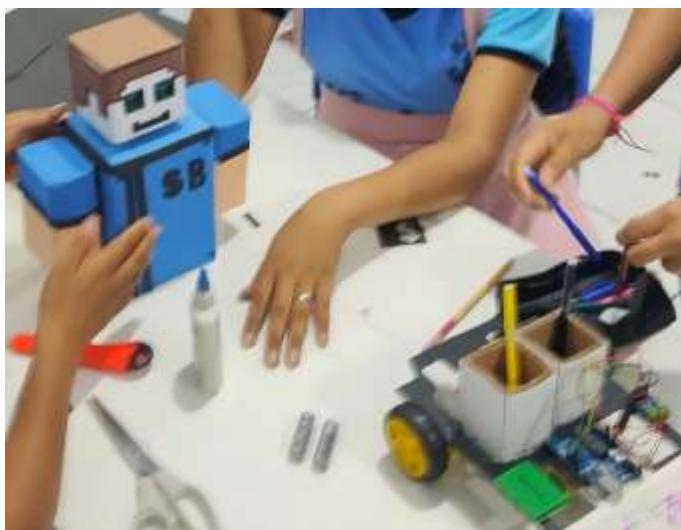


Fonte: Print do software S4A.

6 – JUNÇÃO DAS PARTES ROBÓTICAS E PROGRAMAÇÃO INTEGRAL.

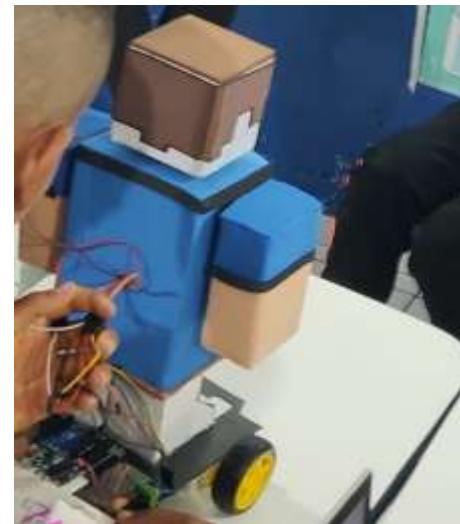
Com a base direcional funcionando, passa-se a acoplagem de todos os protótipos construídos, figura 38, com observação à saída dos fios pela parte de trás do robô, conforme figura 39.

Figura 38 – Junção dos protótipos construídos.



Fonte: Arquivo pessoal.

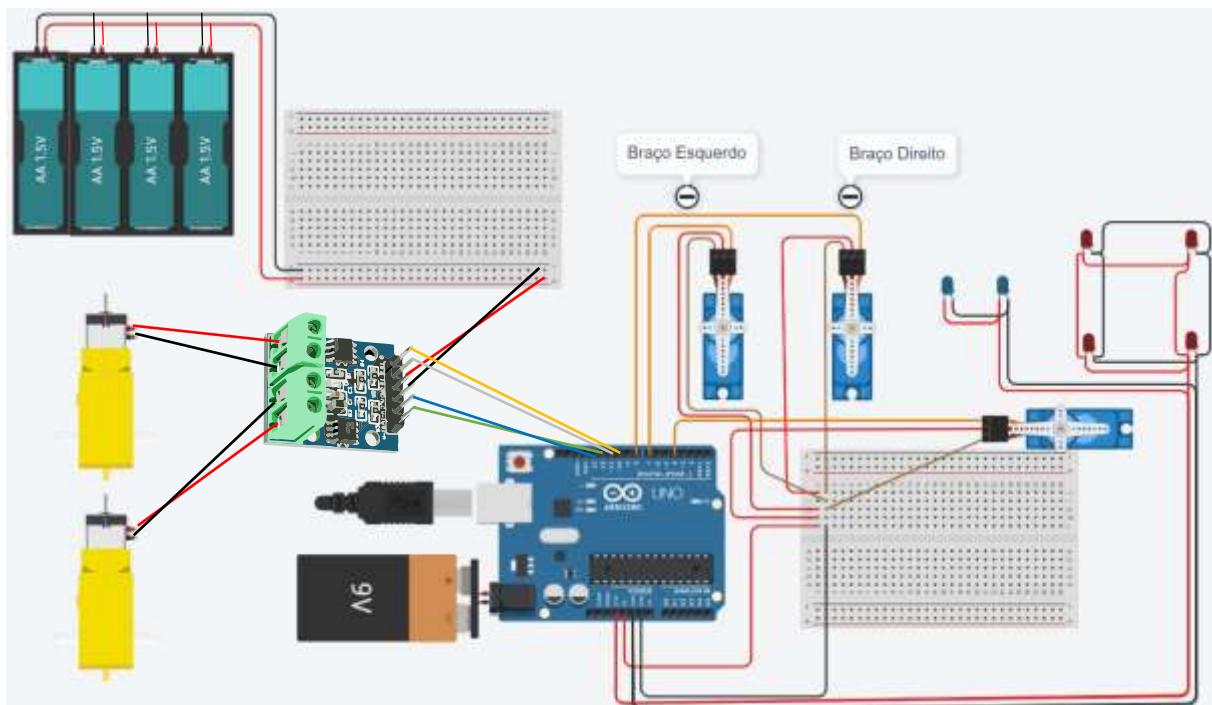
Figura 39 – Saída dos fios pela parte de trás do tronco robótico.



Fonte: Arquivo pessoal.

Com o robô montado, passa-se a organização dos jumpers na placa Arduino e protoboard, pelas quais, para que não haja falhas de potência no funcionamento integral dos servomotores e motores DC's, adicionou-se um circuito de 04 pilhas alcalinas, 1,5v cada, para o acionamento dos motores DC's, figura 40.

Figura 40 – Circuito integral do robô.

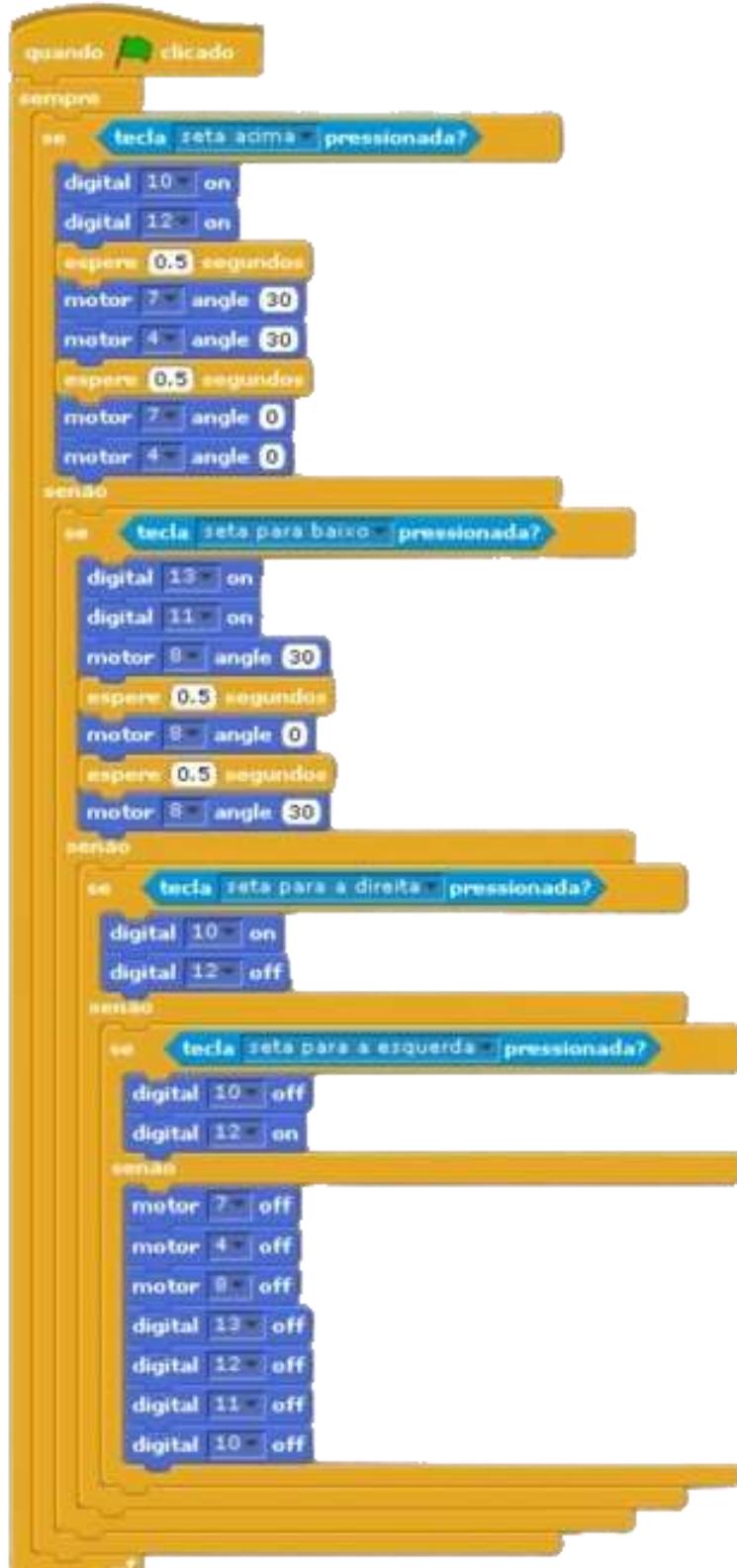


Fonte: Simulação realizada no software Tinkercad.

NOTA: Observe que as saídas de 5V e GND da placa arduino foram transferidas para a protoboard afim de ligar os servomotores, assim como, a saída de 6V das pilhas alcalinas para alimentar os motores DCs.

Com as devidas conexões realizadas, passa-se à junção das sintaxes, a qual será iniciada a partir da programação da base direcional, respeitando cada porta serial dos motores e servomotores, conforme figura 41.

Figura 41 – Sintaxe da programação integral do robô.



Fonte: Sintaxe do software S4A.

Com a sintaxe carregada à placa controladora, o robô deverá:

- Estar com os leds todos ligados;
- Movimentar-se nas 04 direções (frente, trás, direita e esquerda);
- Ao mover-se para frente, movimentar os braços;
- Ao mover-se para trás, movimentar a cabeça;

Robô Direcional.



Fonte: Arquivo pessoal.

8 – POSSÍVEIS ERROS.

- Jumpers mal conectados.
- Falta de potência pelo não uso de baterias.
- Deslizamento das palhetas (servomotor), por não estarem fixas corretamente.
- Entrelaçamento dos fios eletrônicos, gerando desconexões durante os movimentos angulares.
- Erro no posicionamento dos fios na Ponte H, gerando movimentos em direções contrárias às programadas.
- Dimensionamento dos protótipos construídos, o robô deverá ser proporcional aos movimentos e motores utilizados, isto é, um robô muito alto poderá cair ou, ainda, um braço muito grande poderá ter dificuldade para movimentar-se.
- Erros na sintaxe (ferramenta de loop, temporizador, ...).

REFERÊNCIAS

- BRASIL. Ministério da Educação. **Base Nacional Comum Curricular**. Brasília, 2018.
- PAPERT, Seymour. **Logo: Computadores e educação**. São Paulo: Ed. Brasiliense, 1986.
- RESNICK, Mitchel. **Distributed Constructionism**, Northwestern University, International Conference on the learning Sciences, 1996.



Denis Heitor Damasceno da Silva – Possui graduação em Licenciado em Matemática (UEPA) e em Tecnologia em Processamento de Dados (CESUPA), Especialista em Fundamentos da Matemática Elementar pela Universidade do Estado do Pará, Técnico em Óptica e Optometria pelo Instituto Filadélfia – Ribeirão Preto/SP, Mestrando em Ensino da Matemática pela Universidade do Estado do Pará - UEPA. Concursado da Secretaria do Estado de Educação do Pará e da Secretaria Municipal de Educação do Município de Concórdia do Pará. E-mail: profdenisheitor@gmail.com.



Carlos Alberto de Miranda Pinheiro - Doutor em Educação Matemática, Mestre em Educação na linha de Formação de Professores, Especialista em Metodologias para o ensino de Matemática com complementação pedagógica para o exercício da docência no Ensino Superior, Especialista em Educação Matemática, Especialista em Fundamentos de Matemática Elementar, Aperfeiçoamento para o ensino de Matemática no segundo grau. Bacharel em Ciências Náuticas e Licenciado Pleno em Matemática, professor da Universidade do Estado do Pará, da Secretaria de Estado e Educação do Pará e Assessor pedagógica de projetos estratégicos de gestão na Escola de Governança Pública do Estado. Atualmente coordena o curso de licenciatura em Matemática da UEPA. Vice Líder do Grupo de Estudos e Pesquisas em Didática da Matemática e Educação Matemática (CCSE/UEPA). Realiza estudos no campo da Educação Matemática com ênfase: Ensino e Aprendizagem da Análise Combinatória, Formação de Professores que Ensinam Matemática e Metodologias STEAM.



Cinthia Cunha Maradei Pereira - Possui graduação em Licenciatura em Matemática e em Tecnologia em Processamento de Dados, Especialização em Informática Médica, Mestrado em Ciências da Computação e Doutorado em Genética e Biologia Molecular (Bioinformática). Participa do desenvolvimento de tecnologias aplicadas ao ensino de Matemática, Professora da Universidade do Estado do Pará.