

ENSINO DE ROBÓTICA COM S4A – SCRATCH FOR ARDUINO

Lázaro Tiago de Oliveira¹
Leonardo Santos Oliveira²
Kennedy Morais Fernandes³

RESUMO

Um dos múltiplos fatores que dificultam o aprendizado da robótica no meio acadêmico é a falta de ferramentas adequadas para alunos sem conhecimentos prévios em programação de computadores e eletrônica. Na maioria das instituições acadêmicas os cursos de linguagens de programação e robótica tem um foco maior na parte teórica, trazendo assim grandes dificuldades à aprendizagem por parte dos alunos. Desta forma, nesse trabalho objetivou-se o ensino de robótica através da implementação de projetos nos quais o estudante aprende praticando, tornando assim mais fácil a compreensão da lógica de programação e introduzindo conhecimentos de eletrônica com o Arduino. Para a realização deste trabalho foi feito o uso das ferramentas do Arduino e da plataforma do *Scratch for Arduino* (S4A) no ensino de robótica, buscando trazer uma abordagem mais lúdica e leve para o ambiente escolar, visando facilitar o aprendizado, uma vez que o aluno trabalha com uma linguagem de programação simples, interativa e com comandos bem definidos e de fácil entendimento. Como resultado desse processo de ensino, tivemos a construção de um simulador de sinal de trânsito.

Palavras-chave: Arduino; Programação; Robótica; Scratch.

INTRODUÇÃO

Programação e lógica ainda são tabus para muitos estudantes, é notável que um bom número deles possuem dificuldades em aprender a programar computadores. Como os fundamentos de lógica dificilmente são ensinados no ensino básico, os estudantes chegam ao nível superior com

¹ Discente do curso de Bacharelado Interdisciplinar em Ciências da Universidade Federal do Sul da Bahia, sis.lazaro@gmail.com

² Discente do curso de Licenciatura Interdisciplinar em Matemática e Computação da Universidade Federal do Sul da Bahia, leodtwo@gmail.com

³ Professor Doutor em Modelagem Computacional da Universidade Federal do Sul da Bahia, kennedy.fernandes@gmail.com



essa dificuldade em aprender os conceitos de programação, e muitas vezes perdem o gosto pelas ciências exatas por não ter afinidade com algoritmos e lógica de programação. Segundo Mélo et al. (2011, p. 2):

Todavia, no contexto da programação, o que acontece no nível superior, a necessidade de interagir com o computador ainda exige um elevado nível de raciocínio abstrato e de lógica, os quais não são trabalhados na escola durante o ensino fundamental ou médio, além de que podem ser dificultados também pelo uso de linguagens codificadas em baixo nível.

Neste trabalho apresentamos uma proposta para o ensino de programação e robótica, de forma mais simples e prática utilizando a ferramenta visual S4A - *Scratch for Arduino*, fazendo com que o aprendizado se torne algo prazeroso.

METODOLOGIA

Na elaboração das aplicações foi utilizada a plataforma *Scratch for Arduino* (S4A, 2016), que é baseada na plataforma original *Scratch*. O *Scratch* é uma suíte de programação feita pelo MIT (*Massachusetts Institute of Technology*) e é disponibilizada gratuitamente aos usuários. A programação no *Scratch* é feita em blocos de simples manuseio e de fácil entendimento, onde o usuário simplesmente clica e arrasta cada comando, formando assim os blocos que definem uma rotina ou uma função (SCRATCH, 2016). O S4A foi desenvolvido por colaboradores para a integração com os componentes do Arduino, utilizando suas entradas digitais e analógicas (S4A, 2016). O ambiente da plataforma S4A pode ser visto na Figura 1.

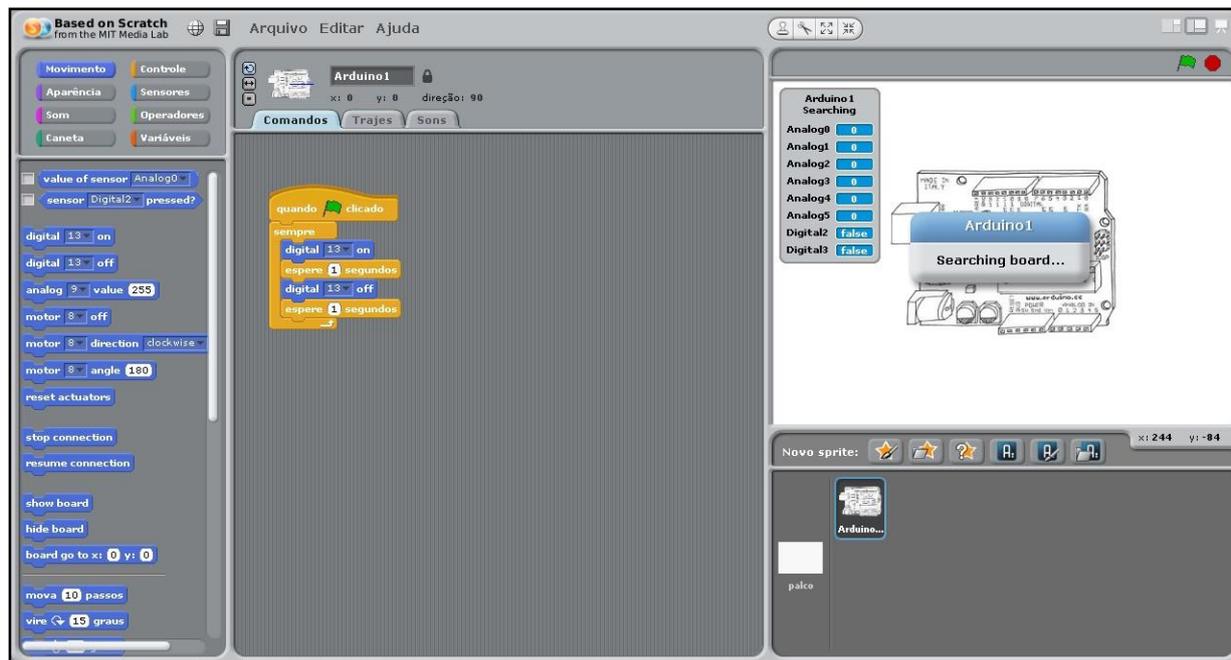


Figura 1 – Ambiente da plataforma S4A.

O Arduino é uma plataforma física de computação de código aberto baseado numa simples placa microcontroladora e um ambiente de desenvolvimento para escrever o código para a placa. Uma placa Arduino é capaz de processar sinais digitais e analógicos através de componentes eletrônicos como botões, *LEDs*, sensores de presença, de luminosidade, entre outros (ARDUINO, 2016).

A plataforma própria do Arduino usa uma linguagem técnica que faz a conexão direta com a máquina, se utilizando de sintaxe padrão, como nas linguagens clássicas de programação. Na Figura 2 é possível ver o esquema de ligação desses componentes. A finalidade deste esquema é fazer um *LED* acender e apagar. Para a implementação deste projeto utilizamos uma placa com o Arduino UNO (ARDUINO, 2016), uma *protoboard* (placa com furos e conexões condutoras), dois *jumpers* (fios) para a ligação e um resistor de 220 ohms para a queda de tensão, pois a porta digital tem uma tensão de 5V e o LED vermelho trabalha numa faixa de tensão de 1,8 V a 2,2 V (KINOPF et al., 2013).

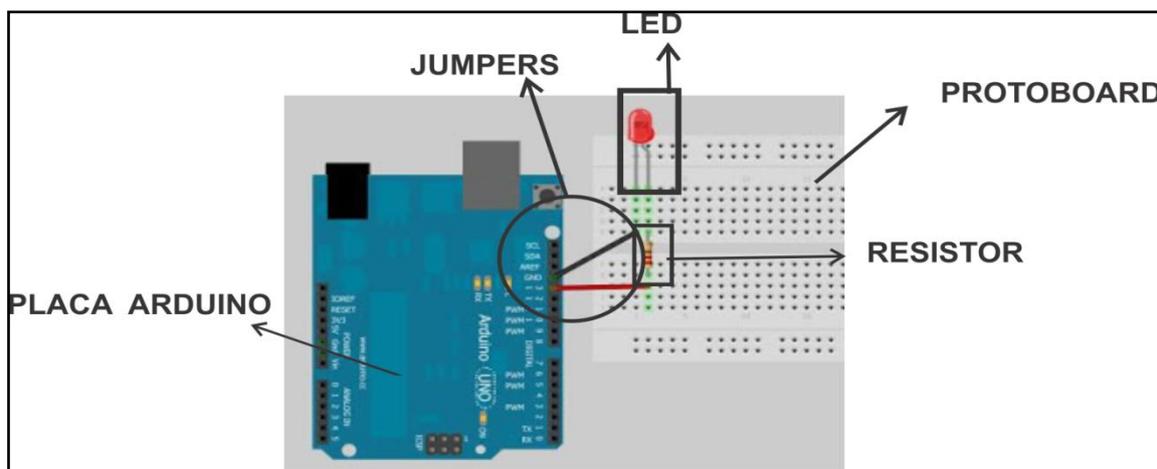


Figura 2 – Esquema de ligação dos componentes do Arduino.

Na *protoboard* há vários furos que correspondem a trilhas interligadas, sendo que as trilhas das extremidades (abaixo e acima) são interligadas na horizontal e as trilhas do meio são interligadas na vertical. O *LED* é ligado de forma que o lado positivo (extremidade maior do *LED*) esteja na mesma trilha de um dos lados do resistor. O resistor, por sua vez, tem o seu outro lado conectado por um *jumper* à placa Arduino em uma de suas saídas digitais, neste projeto foi utilizado a porta digital 13. O lado negativo do *LED* é interligado à entrada GND da placa, onde é o “aterramento” do circuito.

Para acender e apagar o *LED* é preciso programar e estabelecer os comandos. A programação na plataforma do Arduino é mostrada na Figura 3. O código é dividido em duas funções: *setup* e *loop*. O *setup* é a função onde se definem as entradas e saídas que serão utilizadas. Para definir a porta de saída do *LED* foi utilizado o comando *pinMode*, associando-o à porta digital 13. A função *loop* armazena os comandos principais e repete-se continuamente dando dinâmica ao programa e controlando ativamente a placa Arduino. O comando *digitalWrite* estabelece o estado do *LED*, *HIGH* para ligá-lo e *LOW* para desligá-lo, e o comando *delay* define o tempo de espera entre um comando e outro. Sendo assim, o *LED* acende e apaga por um segundo infinitamente. Na Figura 4, a mesma programação foi feita, porém na plataforma S4A. utilizando blocos de comando. Foi utilizado o “sempre” para repetir as ações, e, dentro desse comando, os comandos *digital on* e *digital off* seguidos de “espere” definido em um segundo. Esta simples combinação de blocos de comandos tem a mesma função das linhas de código utilizadas na plataforma do Arduino.

```

void setup()
{
  // Configura o pino d13 como saída
  pinMode(13, OUTPUT);
}

void loop()
{
  // Configura o pino 13 como HIGH
  digitalWrite(13, HIGH);
  // Espera 1000 ms (1 segundo)
  delay(1000);
  // Configura o pino 13 como LOW
  digitalWrite(13, LOW);
  // Espera 1000 ms (1 segundo)
  delay(1000);
}

```

Figura 3 – Programação na plataforma Arduino.



Figura 4 – Programação no ambiente do S4A.

RESULTADOS E DISCUSSÕES

O ensino de robótica com a plataforma Arduino foi eficaz, pois além de atrativo é divertido. As tecnologias novas não poderiam ser indiferentes a nenhum professor, por modificarem as maneiras de viver, de se divertir, de se informar, de trabalhar e de pensar. (PERRENOUD, 2000, p.138). Nesta seção é apresentada a implementação de um simulador de semáforo criado na plataforma S4A (Figura 5) junto com o esquema de ligação na placa Arduino (Figura 6).

Na Figura 5 temos os blocos de comandos implementados no S4A. Iniciamos a porta digital 11 onde está ligado o *LED* verde, digital 12 com o *LED* amarelo e a digital 13 com o *LED* vermelho. O estado inicial de todas as portas é *off*, ou seja, desligadas. Após definir o estado inicial de cada porta digital tem-se um *loop* infinito com o comando “sempre”. Este comando estabelece que quando um *LED* acender, os demais deverão se apagar. Iniciamos o processo de repetição ascendendo o *LED* verde (digital 11 *on*), seguido do comando “espere” com tempo de 5 segundos, após esse tempo o *LED* verde desliga (digital 11 *off*) e o *LED* amarelo liga (digital 12 *on*). O mesmo processo é realizado para ascender os *LEDs* amarelo e vermelho.



Figura 5 – Programação no ambiente do *Scratch*.

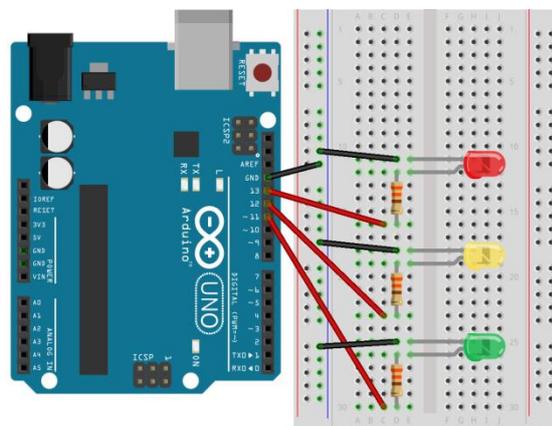


Figura 6 –Esquema de Ligação do simulador de um semáforo.

CONCLUSÃO

Neste trabalho foi proposta uma metodologia para o ensino de robótica através da implementação de projetos, onde o estudante aprende praticando. A compreensão de lógica de programação foi abordada introduzindo conhecimentos de eletrônica com o Arduino.

A metodologia apresentada para o ensino de robótica utilizou o S4A - *Scratch for Arduino*, plataforma de simples manuseio e de fácil entendimento. O S4A facilita a aprendizagem por ser uma plataforma visual e interativa. Como resultado desse processo de ensino, a parte final deste trabalho apresenta a construção de um simulador de sinal de trânsito, um projeto simples que aborda comandos básicos de programação que fazem parte do cotidiano dos alunos.



REFERÊNCIAS

ARDUINO. **What is Arduino?** Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>.

Acesso em 31 de agosto de 2016.

KINOPF, Bruno Silva; ROCHA, Cauã Barneze; CHOCIAY, Lucas; CABRAL, Mariana F.

Machado; ZALESKI, Rafael Henrique. **Sistema de Identificação de Participantes em Postos de**

Controle de Corridas de Orientação. 2013. 34 f. Relatório Final (Oficina de Integração 2) -

Engenharia de Computação, Universidade Tecnológica Federal do Paraná. Curitiba, 2013.

MELO, Francisco Édson Nogueira; CUNHA, Raimundo Ricardo Matos; SCOLARO, Dyonad

Renan; CAMPOS, Jhonatan Luiz. **Do Scratch ao Arduino:** Uma Proposta para o Ensino

Introdutório de Programação para Cursos Superiores de Tecnologia, XXXIX Congresso Brasileiro de Educação em Engenharia, Blumenau, out. 2011.

PERRENOUD, F. **Dez novas competências para ensinar.** Porto Alegre: Artmed, 2000.

S4A. **About S4A.** Disponível em: <<http://s4a.cat>>. Acesso em 30 de agosto de 2016.

SCRATCH. **Acerca do Scratch.** Disponível em: <<https://scratch.mit.edu/about>>. Acesso em 30 de agosto de 2016.