

ENSINO DE MATEMÁTICA ATRAVÉS DO ARDUINO

Porque a placa eletrônica Arduino deve ser inserida no contexto Escolar das aulas de Matemática

Thiago Espindola Cury¹
Daniela Rohan Hirschmann²

Resumo

Atualmente o grande problema encontrado por professores em um contexto geral é a falta de interesse dos alunos para o aprendizado de disciplinas exatas, como é o caso da Matemática. Através do construcionismo, pode ser levado o conhecimento para o aluno através do arduino em conjunto com outras linguagens como o Scratch. Com isso o aluno poderá vivenciar ou pelo menos ter contato visual com a tecnologia que permite a aproximação do que é abstrato, calculado, exato, com luzes piscando, cores, ações que podem acontecer ou sofrer através da interação do aluno com o equipamento. O aluno poderá trazer a matemática e sua exatidão para o seu mundo observando, aprendendo e construindo aplicações reais através dessa placa eletrônica denominada Arduino. O impacto da abstração da matemática e suas funções podem ser minimizados com aulas diferenciadas através de oficinas, onde o aluno possa interagir e construir seu conhecimento baseado nesta ideologia.

Palavras Chave: Arduino, Scratch, Matemática, Construcionismo, Construtivismo

DO MATH THROUGH TEACH ARDUINO Because the Arduino be inserted classrooms give no context School Mathematics

Abstract

Currently the major problem finding by teachers in a general context is the unwillingness of students for learning the exact disciplines, such as mathematics. Through constructionism can be brought knowledge to the student through the arduino in conjunction with other languages such as Scratch. With that the student may experience or at least have visual contact with the technology that allows approximation of what is abstract, calculated accurate with flashing lights, colors, actions that may happen or suffer through student interaction with the equipment. The student may bring mathematics and its accuracy to your world watching, learning and building real applications through this electronic board called Arduino. The impact of abstraction of mathematics and its functions can be minimized with differentiated lessons through workshops, where students can interact and build your knowledge through this ideology.

Keywords: Arduino, Scratch, Mathematic, Constructionism, Constructivism

1 INTRODUÇÃO

¹ Analista e Desenvolvedor de Sistemas. QI Escolas e Faculdades. e-mail: thiagocury@gmail.com

² Mestre em Educação. IERGS/UNIASSELVI. e-mail: danirh21@yahoo.com.br

A matemática é uma disciplina temida por todos e odiada por muitos, isso devido ao fato dos PCNs focarem o ensino da matemática através de fórmulas e cálculos. Atualmente, com o avanço da tecnologia e da informática os professores podem fazer uso de aulas mais dinâmicas que auxiliem os alunos no seu ensinamento empírico, fazendo com que o aluno consiga realizar análises, conclusões, e assim por diante.

Assim como a maioria das grandes áreas, a matemática está repleta de ferramentas, sejam elas softwares ou hardwares para o seu ensino. Alguns softwares já são conhecidos no meio matemático por professores, como softwares para geometria, como: GeoGebra, Gambol, Dr. Geo, entre outros. Ou ainda softwares para álgebra computacional, como: Fermat, FriCAS.

Esse artigo contextualizará sobre o que é, como e onde a placa eletrônica Arduino pode auxiliar o professor de Matemática para suas práticas pedagógicas, auxiliando de forma consistente os alunos no entendimento do conteúdo programático. Pode facilitar o processo de ensino e aprendizagem e melhoria da absorção da lógica matemática por parte dos alunos, tornando assim o ensino de matemática mais significativo.

Através do Arduino os alunos serão instigados, incentivados a procurar e entender como a matemática está diretamente envolvida com o desenvolvimento, seja ele para robôs, jogos, aplicações comerciais, automação residencial, entre outros. Isso contribuirá diretamente para o desenvolvimento cognitivo, autônomo, crítico e consciente dos alunos. Assim, a instituição de Ensino poderá, a partir desse artigo, adaptar a maneira de construir o conhecimento com os alunos através de métodos e soluções mais específicas com a utilização do Arduino.

A ideia desse artigo é elucidar o que é, onde e como o Arduino pode auxiliar na prática pedagógica no ensino de Matemática. Segundo Moran (2011):

Os professores de um modo geral não estão prontos para a multimídia. Ele afirma que eles sentem o descompasso do domínio das tecnologias e muitos têm medo de revelar suas dificuldades, e com isso não estão preparados para experimentar com segurança. Portanto, os professores devem procurar a qualificação para aproveitar os recursos tecnológicos de modo a otimizar o processo de ensino e aprendizagem.

Essa demonstração abrirá uma nova ideia de proposta de ensino aprendizagem com a utilização de ferramentas pedagógicas que auxiliem os alunos no entendimento da disciplina de matemática no seu contexto geral. Com isso, os professores que o lerem poderão refletir sobre as suas práticas pedagógicas atuais no ensino da matemática, talvez tornando a sua prática mais desafiadora e o ensino de seus alunos menos abstrato, fragmentado, fazendo assim com que os mesmos possam ter um aprendizado significativo e mais intuitivo, gerando interesse em buscar seu próprio aprendizado.

Com essa ferramenta o aluno poderá construir seu conhecimento segundo seus próprios interesses. Segundo Gardner, são 9 inteligências múltiplas que nós temos, são elas: linguística, lógico-matemática, musical, especial, corporal, interpessoal, intrapessoal, naturalista e existencialista. Todo ser humano tem essas inteligências de forma variada e distinta, cada um com um determinado nível. Gardner acredita que pode-se melhorar o aprendizado do aluno trabalhando essas diferentes inteligências de forma individual ou conjunta, afetando assim diferentes partes do cérebro. Ao trabalhar as inteligências os alunos passam a compreender o equilíbrio entre as mesmas passando a responsabilizar-se pelo próprio aprendizado.

A partir dessa concepção, o professor que revisar e variar suas práticas pedagógicas, conseguirá atingir os alunos, fazendo com que cada um consiga aprimorar suas inteligências melhorando assim seus níveis de inteligência. O Arduino auxiliará a construção do conhecimento e aprendizagem do aluno, como por exemplo: i) inteligência interpessoal: trabalhando em grupo com o arduino para construir alguma aplicação; ii) inteligência lógico matemática: trabalhando com a linguagem de programação do arduino; iii) inteligência musical: criando aplicações que emitem sons através de frações de tempo.

Os alunos serão estimulados a trabalhar em grupo, desenvolver a criatividade, autoestima, raciocínio lógico e o gosto pela pesquisa. A matemática em prol da construção de robôs no ramo da robótica com certeza desperta qualquer criança/pré adolescente/adolescente a motivação para o aprendizado.

Assim pode ser trabalhado a multidisciplinariedade não só com a matemática mas sim com diversas áreas em conjunto, tais como: álgebra, matemática, física, geografia, português, música, entre outras.

A seção a seguir deste artigo apresenta o desenvolvimento. A seção 2.1 traz uma breve dissertação sobre o Construcionismo, utilizando robótica no auxílio à educação. Na seção 2.2 e suas sub seções será apresentado o arduino e sua história, bem como a linguagem de programação utilizada, destacando seus componentes e as funções matemáticas. Também apresenta a linguagem de programação Scratch, finalizando com as possibilidades de utilização do arduino nas aulas de matemática no contexto geral. As seções 3 e 4 apresentam a conclusão e as referências, respectivamente.

2 DESENVOLVIMENTO

Atualmente o aluno chega na Escola, seja pública ou privada, com conhecimento adquirido sobre informática, tablets, smartphones, netbooks, ultrabooks, porém o mesmo não sabe a extensão de seu conhecimento.

Muitos alunos utilizam a informática de forma errada, acessando redes sociais e conversando com seus amigos, grupos sociais, players de jogos online, entre outros. Não que isso seja errado, entretanto o mesmo aluno com esse diferencial poderia estar utilizando esse conhecimento adquirido para aprender, ler, estudar sobre outros temas que realmente serão de interesse do mesmo para o futuro.

É nesse momento que entra o papel do professor, lembrando que o mesmo tem o conhecimento sobre o currículo e a estrutura disciplinar de tal matéria. No entanto, o mesmo pode levar o conhecimento para o aluno através de atividades que utilizam a tecnologia, como o computador, arduino, linguagens de programação, softwares específicos, entre outros, para modificar as aulas tradicionais transformando-as através de atividades que façam uso destas ferramentas de aprendizagem.

O aluno não se deu conta que pode escolher a maneira que ele quer aprender sobre determinado assunto, todavia o professor pode reforçar a aprendizagem tradicional através de um ambiente inovador, com o objetivo de auxiliar o mesmo a construir seu conhecimento incorporando essas ferramentas.

2.1 CONSTRUCIONISMO X CONSTRUTIVISMO

2.1.1 Construtivismo

O construtivismo é uma teoria desenvolvida por Jean Piaget, onde idealiza-se que o conhecimento do indivíduo (seja um aluno ou não) é construído de dentro para fora através da interação com ambientes naturais de interação social. Essa perspectiva ressalta que através do processo de construção e reconstrução de estruturas cognitivas se constrói o conhecimento. Para esse autor, o conhecimento

não pode ser concebido como algo predeterminado nem nas estruturas internas do sujeito, porquanto estas resultam de uma construção efetiva e contínua, nem nas características preexistentes do objeto, uma vez que elas só são conhecidas graças à mediação necessárias dessas estruturas, e que essas, ao enquadrá-las, enriquecem-nas.

(PIAGET, 2007, p.1)

2.1.2 Construcionismo

O construcionismo traça uma linha paralela ao construtivismo, onde o aluno é responsável pelo seu próprio aprendizado, deixando de lado a perspectiva que o professor é o ser humano que detém o conhecimento. Nessa ideologia o aluno é responsável pelo seu próprio aprendizado, isso através de seus próprios interesses, construído a partir do cotidiano, com o auxílio do computador. Com isso o aluno pode utilizar o computador para construir seu conhecimento através de coisas que façam sentido para ele. Valente (2014) resume:

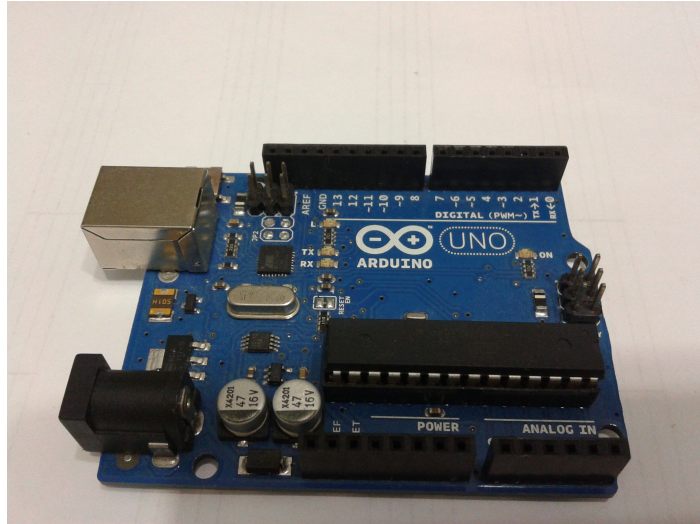
Na noção de construcionismo de Papert, existem duas ideias que contribuem para que esse tipo de construção do conhecimento seja diferente do construtivismo de Piaget. Primeiro, o aprendiz constrói alguma coisa, ou seja, é o aprendizado por meio do fazer, do "colocar a mão na massa". Segundo, o fato de o aprendiz estar construindo algo do seu interesse e para o qual ele está bastante motivado. O envolvimento afetivo torna a aprendizagem mais significativa.

Então, através da placa eletrônica arduino e um computador o aluno pode ser responsável pelo seu próprio conhecimento, buscando o aprendizado nessa perfeita união, fazendo o aprendizado por meio do faça você mesmo, motivando o mesmo através do construcionismo e do seu próprio envolvimento, tornando a sua aprendizagem muito mais significativa.

2.2 O QUE É O ARDUINO?

O Arduino é uma placa eletrônica de prototipagem com microcontrolador utilizada para criar projetos variados de forma independente. O Arduino, que pode ser visualizado na Figura 1, é um conjunto de hardware e software livre, ou seja, pode ser reproduzido e alterado sem problemas, a única observação que deve ser feita é que o nome Arduino é patenteado.

Figura 1 - Arduino



Fonte: produzido pelo autor, 2014

Para desenvolver os códigos é preciso fazer download do IDE (Integrated Development Environment, ou Ambiente de Desenvolvimento Integrado) do Arduino, ou seja, o software que auxilia a escrita dos códigos.

Em suma, o Arduino é um computador “pequeno” capaz de processar entradas e saídas através da própria porta serial do computador onde o arduino está sendo programado ou através de sensores, atuadores conectados a placa de forma externa com o auxílio de uma breadboard.

A maior vantagem do Arduino sobre outras plataformas de desenvolvimento de microcontroladores é a facilidade de sua utilização; pessoas que não são da área técnica podem, rapidamente, aprender o básico e criar seus próprios projetos em um intervalo de tempo relativamente curto. (MCROBERTS, 2011, pág. 3)

O primeiro projeto que é ensinado em qualquer livro ou site relacionado ao arduino é o projeto que consiste em acender e apagar um LED (Diodo emissor de luz – popular luzinha) por uma determinada fração de tempo. O LED pode ser visualizando no anexo I. Esse tempo geralmente é marcado através de uma função onde pode ser aplicado o tempo que o led deve ficar aceso ou apagado através de milissegundos. A figura 2 mostra ao ambiente de programação do Arduino.

Figura 2 - Ambiente de programação

```
sketch_sep10a.s
//Toda essa Area branca chamamos de "Area de Programação"
void setup(){
  //Aqui inicializamos as variaveis
}
void loop(){
  /* Essa e a função principal do arduino. Tudo que for programado nessa função
  sera executado a partir do momento que o controlador(arduino) for ligado.
  Podemos pensar que essa função "tem o comportamento similar" ao de uma
  instrução de "while" infinito. */
}
```

Fonte: Autor, 2014.

Na barra de botões tem-se:

- verify - visto – compila/verifica o software digitado
- upload - seta direita - grava o programa na placa
- New - arquivo novo – cria um novo programado
- Open - seta para cima – abrir programado
- Save - seta para baixo – salvar programado
- Serial Monitor - lupa a direita – monitor serial onde visualizaremos as e/s Área de Programação

A área de programação é dividida em 2 partes (funções), são elas:

- Função Setup: a função setup onde iniciamos as variáveis, placa e do programa. Essa função é executada uma única vez quando a placa é ligada ou reiniciada.
- Função Loop: o void loop é a função principal do arduino, ou seja, tudo que for programado dentro dessa função será executado a partir do momento que o componente eletrônico for ligado. Essa função é executada ininterruptamente enquanto a placa estiver ligada.

Antes das funções pode ser encontrada a área branca no início onde deve ser declarado as variáveis.

2.2.1 Linguagem de programação

O microcontrolador é programado com a linguagem de programação Arduino, baseada na linguagem Wiring, e o ambiente de desenvolvimento Arduino, baseado no ambiente Processing. Utilizar o Arduino separadamente ou em conjunto com o Scratch vem ao encontro com o paradigma construcionista, assim como a perspectiva de Papert através da linguagem Logo. A tecnologia pode e deve ser utilizada e canalizada a serviço da educação como elemento auxiliar conforme os ambientes que os professores podem criar através de sua utilização.

A Linguagem Logo (PAPERT, 1980) é uma linguagem de programação criada para ensinar crianças a programar, e a partir daí construir conhecimento em várias áreas, principalmente a matemática. Não é atoa que os alunos da 4ª série do ensino fundamental estão aprendendo linguagens de programação e robótica em algumas instituições particulares de ensino.

2.2.2 Operadores

No arduino pode ser utilizado operadores aritméticos, lógicos e de comparação. Com os operadores podem ser realizados cálculos matemáticos, testes através de comparações como: maior, menor, igual ou até mesmo testes lógicos com mais de uma condição. A seguir os quadros 1, 2 e 3 demonstram os operadores aritméticos, operadores de comparação e lógicos, respectivamente:

Quadro 1 - Operadores aritméticos

Operador	Significado
+	Adição
-	Subtração
*	Divisão
/	Multiplicação

Fonte: elaborado pelo autor, 2014

Quadro 2 - Operadores de comparação

Operador	Significado
>	Maior
<	Menor
==	Igual
>=	Maior ou igual
<=	Menor ou igual
!=	Não igual

Fonte: elaborado pelo autor, 2014

Quadro 3 - Operadores Lógicos

Operador	Significado
&&	E (and)
	Ou (or)
!	Não (not)

Fonte: elaborado pelo autor, 2014

2.2.3 Funções seriais

A partir da classe “Serial” consegue-se utilizar algumas funções essenciais para o desenvolvimento de aplicações com o Arduino. Essa classe contém funções importantes para realizar a entrada de dados a partir da interação com sensors externos ou até mesmo pelo usuário através do terminal serial. As principais funções estão descritas no quadro.

Quadro 4 – Funções seriais

Função	Significado
Serial.begin(taxa Comunicação)	Função utilizada para abrir a conexão serial com determinada taxa de transmissão/recepção.
Serial.end()	Função desabilita a conexão serial
Serial.read()	Função utilizada para ler o primeiro byte que está no buffer da porta serial.
Serial.print(); Serial.println();	As funções print e println são utilizadas para imprimir valores na porta serial. Esses valores podem ser visualizados a partir do terminal. A função println imprime e automaticamente muda para a próxima linha, enquanto a função print apenas imprime sem mudar para a próxima linha.

Fonte: Arduino, 2014

A realização de impressões simples na tela do monitor serial do Arduino pode ser

realizada através do comando `Serial.println(“”)`. Para visualizar a saída que será impressa é obrigatória a declaração do comando `Serial.begin(velocidade)` dentro da função “setup”. A Figura 3 ilustra a execução dos comandos supracitados.

Figura 3 – comandos begin e println



```
File Edit Sketch Tools Help
sketch_sep29a 5

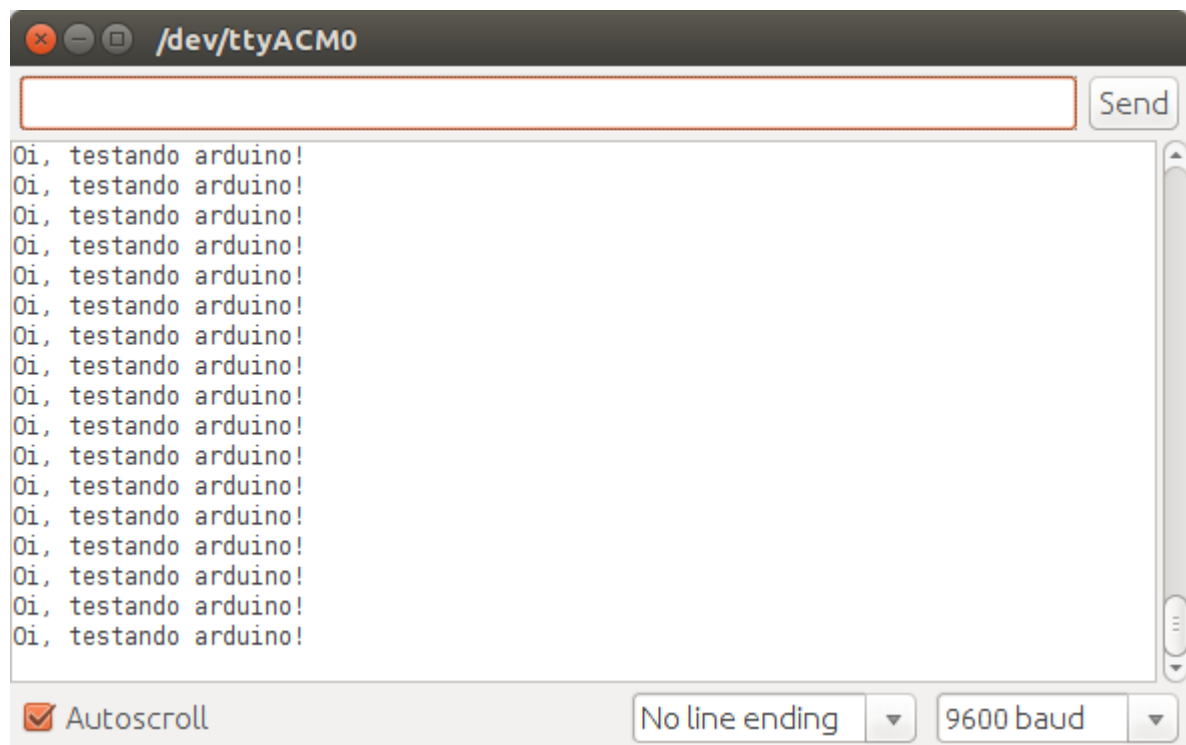
void setup(){
  Serial.begin(9600);
}

void loop(){
  Serial.println("Oi, testando arduino!");
}
```

Fonte: produzido pelo autor, 2014

Como não foi delimitado um tempo no código da Figura 3, as informações serão impressas no monitor Serial conforme Figura 4 ininterruptamente enquanto a placa eletrônica estiver ligada.

Figura 4 - resultado



```
/dev/ttyACM0
Send
Oi, testando arduino!
Oi, testando arduino!
Oi, testando arduino!
Oi, testando arduino!
Oi, testando arduino!
Oi, testando arduino!
Oi, testando arduino!
Oi, testando arduino!
Oi, testando arduino!
Oi, testando arduino!
Oi, testando arduino!
Oi, testando arduino!
Oi, testando arduino!
Oi, testando arduino!
Oi, testando arduino!
Oi, testando arduino!
Oi, testando arduino!
Oi, testando arduino!
Oi, testando arduino!
Oi, testando arduino!
```

Autoscroll No line ending 9600 baud

Fonte: produzido pelo autor, 2014

2.2.4 Funções Matemáticas e de Tempo

Através das funções matemáticas e de tempo concatenadas com os operadores apresentados anteriormente, a linguagem de programação Arduino torna-se poderosa e extremamente instigante. Pode-se construir aplicações cujo funcionamento avalie frações de tempo para iniciar ou concluir determinada tarefa, ou ainda iniciar ou parar algum motor,

atuador, entre outros.

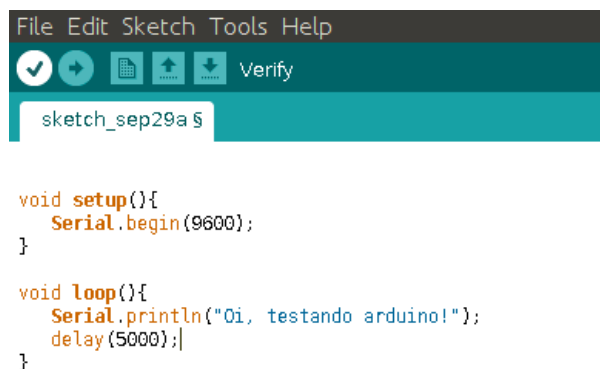
Quadro 5 – Funções matemáticas e de tempo

Função	Utilidade
delay(ms)	pausa o programa por determinado período em milissegundos
delayMicroseconds(us)	pausa o programa por determinado período em microssegundos
millis()	retorna o número de milissegundos desde quando o Arduino começou a executar o programa.
random(min,max)	gera números randômicos entre os limites definidos em “min” e “max”.
abs(x)	retorna o módulo absoluto do número passado como parâmetro.
map(valor, min1, max1, min2, max2)	converte uma faixa de valores para outra faixa. O primeiro parâmetro valor é o valor que será convertido. Os valores min1, max1 são os valores máximos para o valor que foi recebido, já os valores min2 e max2 são os valores correspondentes a nova faixa.

Fonte: Arduino, 2014

Como exemplo de função matemática a Figura 5 ilustra como pode ser visualizado a impressão na tela de texto sendo processado de 5 em 5 segundos. Nesse exemplo foi utilizada a função delay() que indica em milissegundos quanto tempo o controlador deve pausar até executar a próxima instrução.

Figura 5 – função delay



```
File Edit Sketch Tools Help
[Icons] Verify
sketch_sep29a §

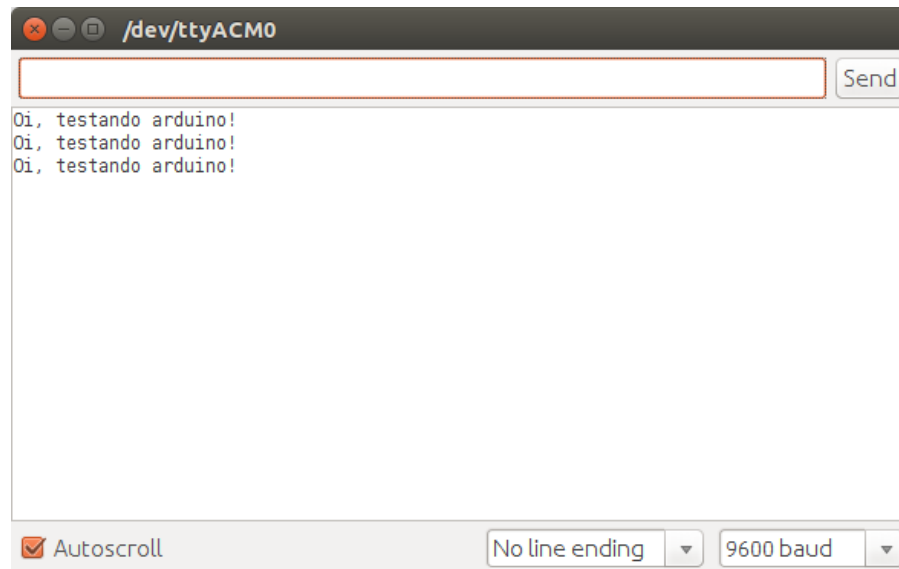
void setup(){
  Serial.begin(9600);
}

void loop(){
  Serial.println("Oi, testando arduino!");
  delay(5000);
}
```

Fonte: produzido pelo autor, 2014

A Figura 6 permite visualizar a execução do código no terminal serial. A Figura demonstra apenas 3 impressões sobre a frase “Oi, testando Arduino!”, para que seja demonstrado o tempo decorrido de 15 segundos, ou seja 15000 milissegundos executados a partir da execução da função delay(5000) três vezes.

Figura 6 - resultado



Fonte: produzido pelo autor, 2014

2.2.5 Declaração de variáveis

Além de todos operadores e funções, o arduino permite a declaração de variáveis e constantes, sendo que essas podem ser:

- Variáveis simples para armazenar somente 1 valor de forma variável;
- Variáveis constantes para armazenar somente 1 valor de forma fixa;
- Variáveis unidimensionais, os conhecidos vetores que permitem armazenar mais de um valor ao mesmo tempo;
- Variáveis bidimensionais, as matrizes que permitem armazenar valores de forma bidimensional.

2.2.6 Pinos

A placa é constituída por 14 pinos digitais e 6 pinos analógicos. Os pinos digitais são numerados de 0 a 13 e os pinos analógicos são identificados de 0 a 5. Os pinos digitais podem ser configurados para detectarem ou transmitirem informações, já os analógicos são utilizados para leitura de sinais analógicos de sensores conectados.

Para configurar os pinos existe uma função chamada `pinMode()` que permite a transmissão de 2 estados, verdadeiro ou falso(1/0) através das palavras OUTPUT ou INPUT.

Para acender um led por exemplo é utilizada outra função denominada `digitalWrite()`, onde são inseridos os valores HIGH ou LOW, sendo HIGH para ligar liberar a tensão, ou seja ligar o LED e LOW para barrar a tensão, desligando o led.

2.2.7 Componentes

O arduino pode ser utilizado para trabalhar de forma individual programando apenas na placa e visualizando o resultado no terminal serial ou pode utilizar componentes como: breadboard, jumpers, joystick, tela de LCD (16x2), sensores diversos, leds, resistores, motores, entre outros. Esses componentes citados foram escolhidos por terem algum tipo de

envolvimento com a matemática, ou seja, podem ser utilizados para demonstrações de coordenadas, frações de tempo, testes.

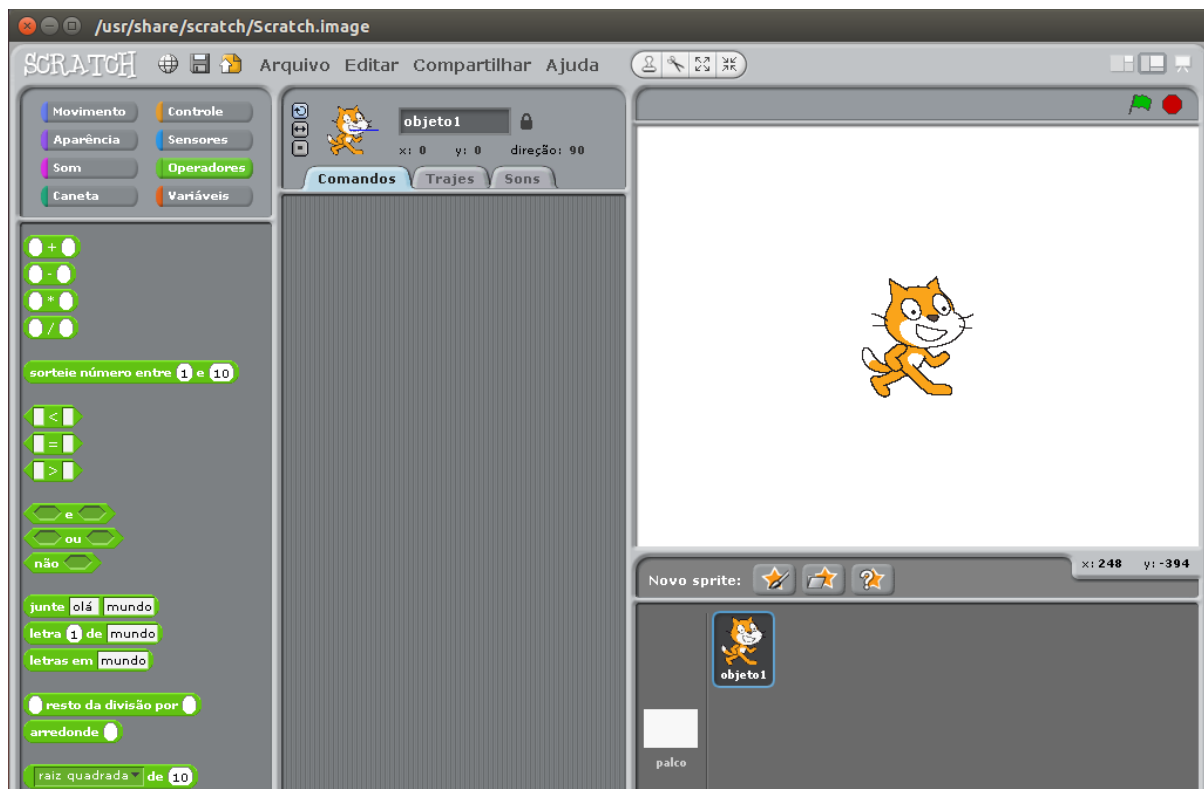
Os jumpers (fios) são utilizados para fazer todas as conexões necessárias juntamente com a protoboard (breadboard). A imagem dos jumpers pode ser visualizada no anexo II. A breadboard por sua vez é uma matriz de contatos para realizar as conexões necessárias sem precisar soldar nenhum fio. A imagem do componente breadboard pode ser visualizada no anexo III.

2.2.8 Scratch

O Scratch é uma linguagem de desenvolvimento através de blocos pré definidos de comandos, onde conseguimos criar aplicações, jogos, utilizando arquivos de media tais como: músicas, sons, gráficos, etc). O intuito do Scratch é aprimorar e facilitar o processo de ensino aprendizagem de programação para crianças, especialmente entre as idades de 8 à 16 anos. O grande diferencial do Scratch é sua linguagem de programação em blocos ser gráfico, ou seja, o aprendiz só precisa arrastar os blocos de programação, criando suas próprias condições, etc. Esse ambiente foi desenvolvido pelo M.I.T., inspirado na linguagem Logo.

O Scratch apresenta três áreas básicas e distintas, são elas: a paleta de funções, a área de código e o cenário de execução do programa contendo o gato, conforme Figura.

Figura 7 – Tela do Scratch



Fonte: produzido pelo autor, 2014

O Scratch permite que os códigos sejam compartilhados no site oficial (<http://scratch.mit.edu/>), onde os softwares podem ser testados, executados através do navegador ou permite que se faça o *download*. Além de efetuar o *download* e testar, os códigos podem ser incorporados em uma página Web.

2.2.9 Softwares Matemáticos

No site oficial do Scratch encontram-se diversos softwares voltados para o ensino da matemática, indiferente do nível de dificuldade, os softwares trabalham desde operações básicas até coordenadas de x e y, funções cálculos complexos, calculadoras, entre outros.

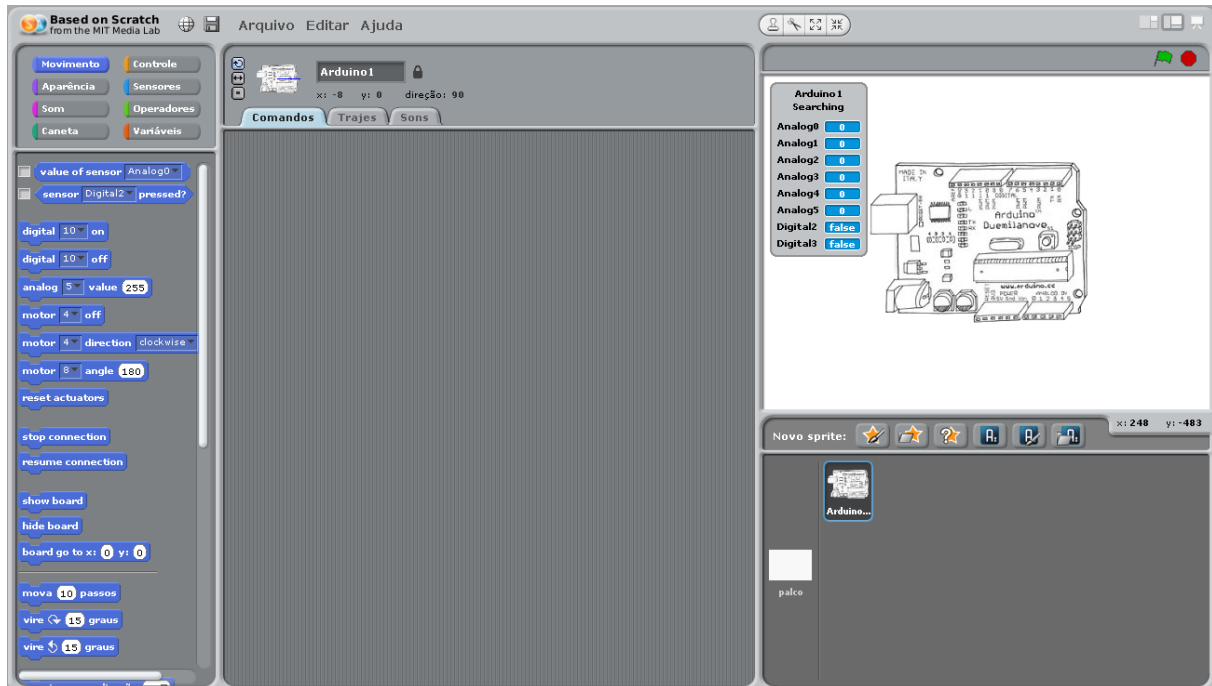
Abaixo segue a lista dos links de alguns desses softwares:

- Corrida de Lesmas, conforme operações aritméticas básicas + - / *: <http://scratch.mit.edu/projects/13853051/>
- Quiz com operações aritméticas básicas + - / *: <http://scratch.mit.edu/projects/12115110/>
- Corrida de carro, com operações aritméticas básicas + - / *: <http://scratch.mit.edu/projects/2377449/>
- Mostra o gato no plano cartesiano, conforme arrata ele mostra as coordenadas de x e y atuais: <http://scratch.mit.edu/projects/685640/>
- Jogo do Yoshi com operações aritméticas básicas + - / *: <http://scratch.mit.edu/projects/1974091/>
- Jogo de Lógica booleana(and e or) com Banana : <http://scratch.mit.edu/projects/761657/>
- Conversor de binário para Decimal : <http://scratch.mit.edu/projects/657082/>

2.2.10 S4A

O S4A (Scratch For Arduino) é uma modificação do software original Scratch que auxilia na utilização da placa eletrônica arduino em conjunto com o Scratch. A partir dessa modificação o S4A fornece novas opções voltadas especificamente para utilização de LEDs, sensores, atuadores conectados ao arduino. Com isso pode-se programar no Arduino a partir do Scratch utilizando os resultados dos sensores e processos que estão acontecendo no Arduino. O usuário pode interagir com o Arduino e o mesmo enviará as informações para o Scratch, este interpretará a leitura dos sensores e executará a programação que foi criada a partir dos blocos no Scratch. Então, através do S4A pode ser integrado os 2 ambientes para que trabalhem de forma conjunta, facilitando a programação de crianças, jovens, adultos, entre outros.

Figura 8 – S4A



Fonte: produzido pelo autor, 2014

2.3 COMO UTILIZAR O ARDUINO NAS AULAS DE MATEMÁTICA

O Arduino pode ser adotado de duas maneiras diferentes, são elas: o professor leva o arduino para ilustrar determinada matéria como, por exemplo, matrizes através de leds. Isso auxiliará os alunos no processo de aprendizagem, porém os mesmos não terão contato com a placa eletrônica.

A segunda maneira consiste na apresentação do Arduino, conceitos básicos de programação e principais componentes. Desta forma, os alunos poderiam construir pequenas aplicações com o intuito de resolver problemas matemáticos através da linguagem de programação. Também poderia ser utilizado para o ensino de matérias mais básicas, como frações. Alguns projetos simples que podemos citar: sistema binário, conversão de bases, utilização das próprias funções básicas do mesmo, cálculos simples através de operadores aritméticos. Entre opções de projetos mais completos pode ser trabalhado com vetores, matrizes, jogo da velha e jogo da memória.

A seguir serão apresentados alguns códigos de projetos pequenos que podem ser úteis nas aulas de matemática. Exemplos básicos de aulas que podem ser aplicados, são:

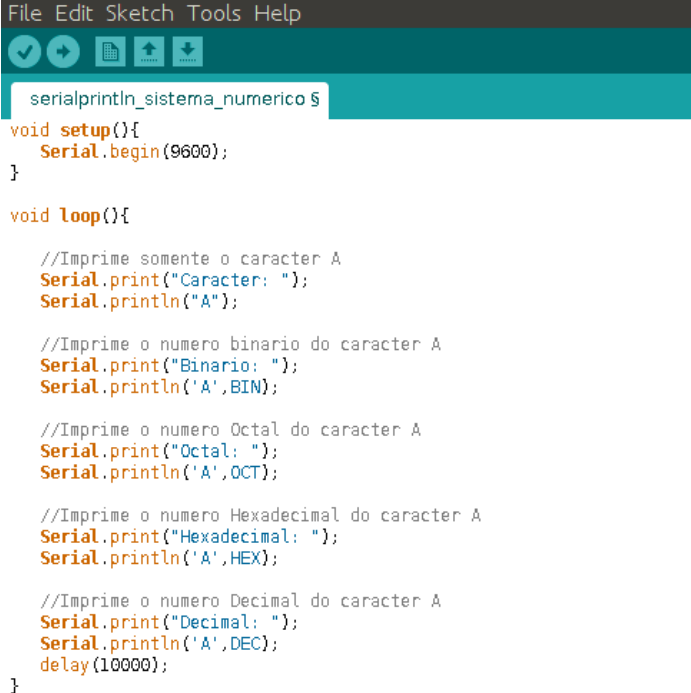
- Sistema numérico;
- Calculadora simples;
- Cálculos como raio de círculo, bháskara, entre outros;
- Led piscando;
- Array de leds Vetores;
- Vetores e matrizes utilizando laço for;
- Matriz com dados fixos;
- Matriz com dados randômicos;

2.3.1 Sistema numérico

Através das funções seriais pode ser obtido o resultado de valores em diversos sistemas numéricos, todavia o professor pode utilizar o Arduino para tirar a prova real ou até mesmo para que os alunos o utilizem para verificar se os seus cálculos estão corretos ou não.

A Figura 9 demonstra o código com as funções seriais sendo utilizadas para imprimir o character “A” nos seguintes sistemas numéricos: binário, octal, hexadecimal e decimal. Já a Figura 10 demonstra a saída para o usuário.

Figura 9 – código com funções seriais



```

File Edit Sketch Tools Help
serialprintln_sistema_numerico S
void setup(){
  Serial.begin(9600);
}

void loop(){

  //Imprime somente o caracter A
  Serial.print("Caracter: ");
  Serial.println("A");

  //Imprime o numero binario do caracter A
  Serial.print("Binario: ");
  Serial.println('A',BIN);

  //Imprime o numero Octal do caracter A
  Serial.print("Octal: ");
  Serial.println('A',OCT);

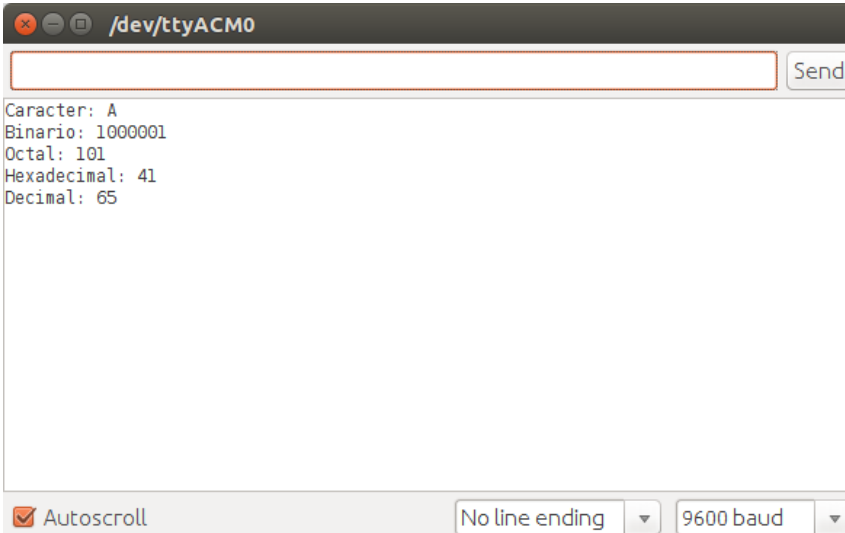
  //Imprime o numero Hexadecimal do caracter A
  Serial.print("Hexadecimal: ");
  Serial.println('A',HEX);

  //Imprime o numero Decimal do caracter A
  Serial.print("Decimal: ");
  Serial.println('A',DEC);
  delay(10000);
}

```

Fonte: produzido pelo autor, 2014

Figura 10 - resultado



```

/dev/ttyACM0
Caracter: A
Binario: 1000001
Octal: 101
Hexadecimal: 41
Decimal: 65

```

Autoscroll No line ending 9600 baud

Fonte: produzido pelo autor, 2014

2.3.2 Calculadora simples

Esse projeto tem como objetivo demonstrar a utilização da matemática e o funcionamento do Arduino sem a utilização de nenhum componente, visualizando a saída direto no terminal serial do próprio Arduino. Com isso o aluno utilizará operadores, funções para realizar cálculos dos mais diversos. A Figura 11 demonstra o código implementado de uma simples calculadora que efetua as 4 operações básicas. Os valores são inseridos de forma fixa nas variáveis direto nas variáveis no software.

Figura 11 - calculadora



```
File Edit Sketch Tools Help
calculadora_basica
//Declaracao de variaveis
float numero1 = 0;
float numero2 = 0;
float resposta = 0;

void setup(){
  //Iniciando serial
  Serial.begin(9600);
}

void loop(){
  //Entrada de dados fixa
  numero1 = 10;
  numero2 = 5;

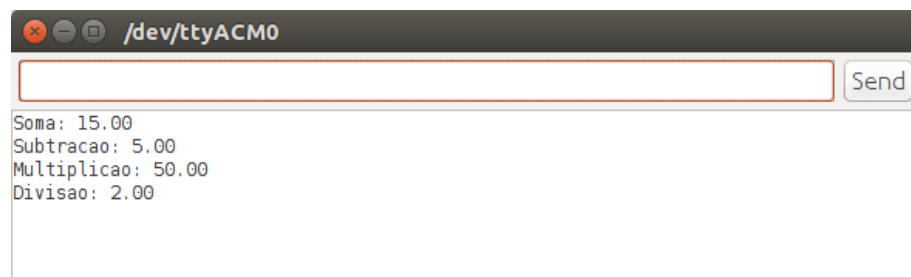
  //Soma:
  Serial.print("Soma: ");
  resposta = numero1 + numero2;
  Serial.println(resposta);
  //Subtracao:
  Serial.print("Subtracao: ");
  resposta = numero1 - numero2;
  Serial.println(resposta);
  //Multiplicacao
  Serial.print("Multiplicacao: ");
  resposta = numero1 * numero2;
  Serial.println(resposta);
  //Divisao
  Serial.print("Divisao: ");
  resposta = numero1 / numero2;
  Serial.println(resposta);

  delay(10000);
}
```

Fonte: produzido pelo autor, 2014

O resultado pode ser visualizado na figura 12.

Figura 12 - resultado



```
/dev/ttyACM0
Soma: 15.00
Subtracao: 5.00
Multiplicacao: 50.00
Divisao: 2.00
```

Fonte: produzido pelo autor, 2014

2.3.3 Cálculo de área do Círculo

Esse projeto tem como objetivo demonstrar a utilização da matemática para ensinar

cálculos que podem parecer simples, entretanto estes podem ser utilizados futuramente para verificar quanto que um robô pode andar, por exemplo. Já que o funcionamento das rodas de um robô se dão a partir do cálculo de raio, círculo, circunferência. Com isso pode ser medido qual será a quantidade de voltas que a roda deverá executar. A Figura 13 demonstra o código implementado para o cálculo de área do círculo.

Figura 13 – área do círculo



```
File Edit Sketch Tools Help
area_do_circulo
//Constante
#define PI 3.14

//Variavel
float area=0;
float raio=0;

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

void loop(){

  //Entrada de dados fixa
  raio = 10;

  //Calculando a area do circulo
  area = PI * (raio*raio);

  //Imprimindo a area do circulo
  Serial.print("A area do circulo e: ");
  Serial.println(area);

  delay(10000);
}
```

Fonte: produzido pelo autor, 2014

O retorno do cálculo pode ser visualizado na Figura 14.

Figura 14 - resultado



```
/dev/ttyACM0
A area do circulo e: 314.16
```

Fonte: produzido pelo autor, 2014

2.3.4 Led piscando

Com os leds podem ser acendidos de forma individual ou conjunta através de testes lógicos, matemáticos ou até mesmos através de frações de tempo com a utilização das funções adequadas. Para acender um led basta inserir o código da Figura 15, ligar os jumpers conforme figura 16. Para esse projeto o led foi ligado a partir da porta digital 2 na placa Arduino. Um resistor foi utilizado para resistir a tensão de 5v reduzindo para 3v. Com isso não tem risco do led queimar. O led pode ou não piscar conforme a programação que é realizada. Caso seja desejado que o mesmo pisque, deve ser utilizada a função matemática delay. Também pode ser solicitada que o led pisque aleatoriamente, para isso basta utilizar a função random. Isso abre portas para o professor de matemática utilizar um led piscando para ensinar frações e cálculos de tempo.

Figura 15 - código led piscando

```
File Edit Sketch Tools Help
_1_led_piscando
//Declaracao de uma variavel led
int led=2;

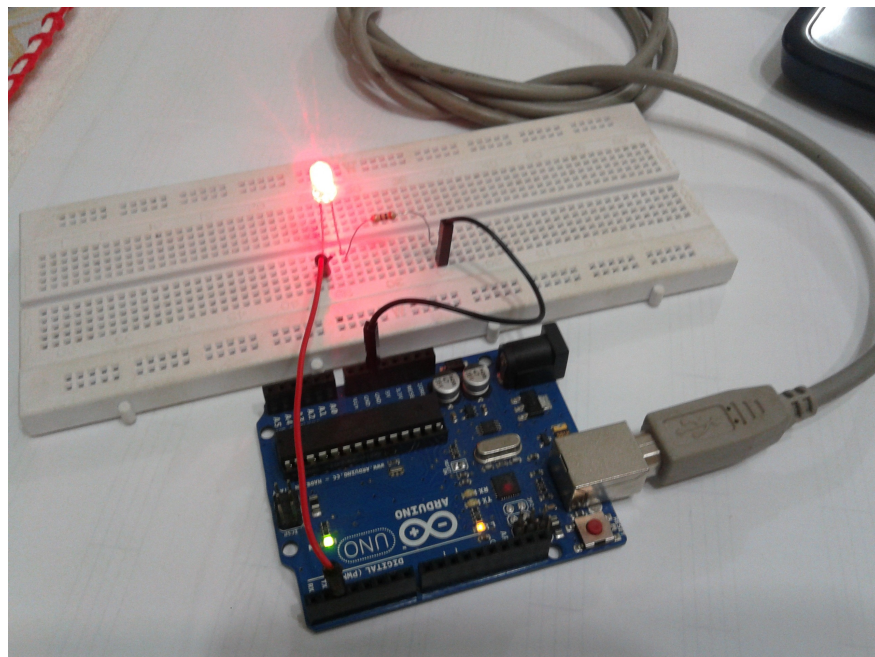
void setup(){
  //Informando que a porta sera utilizada para saida
  pinMode(led,OUTPUT);
}

void loop(){
  //Acende o led durante 1 segundo
  digitalWrite(led,HIGH);
  delay(1000);

  //Apaga o led durante 1 segundo
  digitalWrite(led,LOW);
  delay(1000);
}
```

Fonte: produzido pelo autor, 2014

Figura 16 - led piscando



Fonte: produzido pelo autor, 2014

2.3.5 Array de leds: Vetores

Para a introdução do ensino de vetores pode ser utilizadas variáveis bidimensionais no Arduino. Com isso o valor das portas lógicas podem referenciar leds e pode ser programado para que os mesmos acendam conforme cada posição do vetor. Com isso os alunos perceberão exatamente em qual posição do vetor o código encontra-se naquele momento. A Figura 17 demonstra o código utilizando vetor e a Figura 18 demonstra o funcionamento juntamente com as ligações que foram realizadas para o array de leds.

Figura 17 - arrays



```

File Edit Sketch Tools Help
_3_leds_piscando_vetor 5
//Declaracao de um vetor com tres valores
int led[]={2,3,4};

void setup(){
  //Informando que a porta sera utilizada para saida
  pinMode(led[0],OUTPUT);
  pinMode(led[1],OUTPUT);
  pinMode(led[2],OUTPUT);
}

void loop(){
  //Acende os leds
  digitalWrite(led[0],HIGH);
  digitalWrite(led[1],HIGH);
  digitalWrite(led[2],HIGH);
  delay(1000);

  //Apaga os leds
  digitalWrite(led[0],LOW);
  digitalWrite(led[1],LOW);
  digitalWrite(led[2],LOW);
  delay(1000);
}

```

Fonte: produzido pelo autor, 2014

Figura 18 –arrays de leds



```

File Edit Sketch Tools Help
_3_leds_piscando_vetor 5
//Declaracao de um vetor com tres valores
int led[]={2,3,4};

void setup(){
  //Informando que a porta sera utilizada para saida
  pinMode(led[0],OUTPUT);
  pinMode(led[1],OUTPUT);
  pinMode(led[2],OUTPUT);
}

void loop(){
  //Acende os leds
  digitalWrite(led[0],HIGH);
  digitalWrite(led[1],HIGH);
  digitalWrite(led[2],HIGH);
  delay(1000);

  //Apaga os leds
  digitalWrite(led[0],LOW);
  digitalWrite(led[1],LOW);
  digitalWrite(led[2],LOW);
  delay(1000);
}

```

Fonte: produzido pelo autor, 2014

2.3.6 Vetor utilizando laço for

O próximo projeto é uma melhoria do projeto anterior sobre vetores, entretanto este utilizando a instrução “for” para não precisar repetir três vezes as linhas pinMode() e digitalWrite(). O for é um laço que permite que o código seja executado o número de vezes necessárias, conforme condição. A Figura 19 ilustra como ficou o código após a alteração.

Figura 19 - laço for



```

File Edit Sketch Tools Help
_3_leds_piscando_vetor_for $
//Declaracao de um vetor com tres valores
int led[]={2,3,4};

void setup(){
  //Informando que a porta sera utilizada para saida
  pinMode(led[0],OUTPUT);
  pinMode(led[1],OUTPUT);
  pinMode(led[2],OUTPUT);
}

void loop(){
  //Acende os leds
  for(int i=0 ; i<3 ; i++){
    digitalWrite(led[i],HIGH);
    delay(1000);
  }

  //Apaga os leds
  for(int i=0 ; i<3 ; i++){
    digitalWrite(led[i],LOW);
    delay(1000);
  }
}

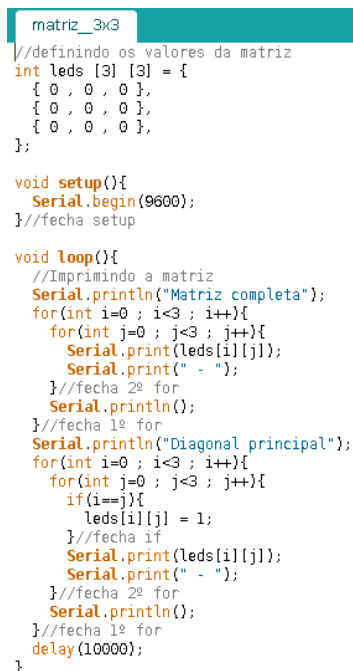
```

Fonte: produzido pelo autor, 2014

2.3.7 Matriz com dados fixos

Cálculos de matrizes podem ser programados no Arduino para serem mostrados no terminal serial ou para serem mostrados através de leds. Para a utilização de matrizes podem ser utilizadas variáveis bidimensionais. As Figuras 20 e 21 demonstram, respectivamente, o código e a visualização no terminal serial. Esse projeto teve como base uma matriz 3x3 que foi preenchida com “0” zeros. A visualização foi dividida em duas partes, são elas: foi visualizada toda a matriz com os valores “0” e a segunda parte foi mostrada apenas a diagonal principal da matriz, substituindo os valores “0” por “1”.

Figura 20 - matriz



```

matriz_3x3
//definindo os valores da matriz
int leds [3] [3] = {
  { 0 , 0 , 0 },
  { 0 , 0 , 0 },
  { 0 , 0 , 0 },
};

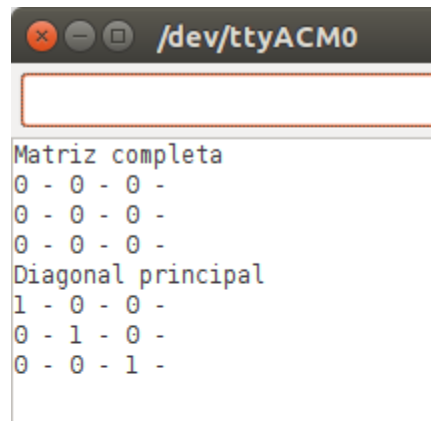
void setup(){
  Serial.begin(9600);
} //fecha setup

void loop(){
  //Imprimindo a matriz
  Serial.println("Matriz completa");
  for(int i=0 ; i<3 ; i++){
    for(int j=0 ; j<3 ; j++){
      Serial.print(leds[i][j]);
      Serial.print(" - ");
    } //fecha 2º for
    Serial.println();
  } //fecha 1º for
  Serial.println("Diagonal principal");
  for(int i=0 ; i<3 ; i++){
    for(int j=0 ; j<3 ; j++){
      if(i==j){
        leds[i][j] = 1;
      } //fecha if
      Serial.print(leds[i][j]);
      Serial.print(" - ");
    } //fecha 2º for
    Serial.println();
  } //fecha 1º for
  delay(10000);
}

```

Fonte: produzido pelo autor, 2014

Figura 21 - resultado



```

Matriz completa
0 - 0 - 0 -
0 - 0 - 0 -
0 - 0 - 0 -
Diagonal principal
1 - 0 - 0 -
0 - 1 - 0 -
0 - 0 - 1 -

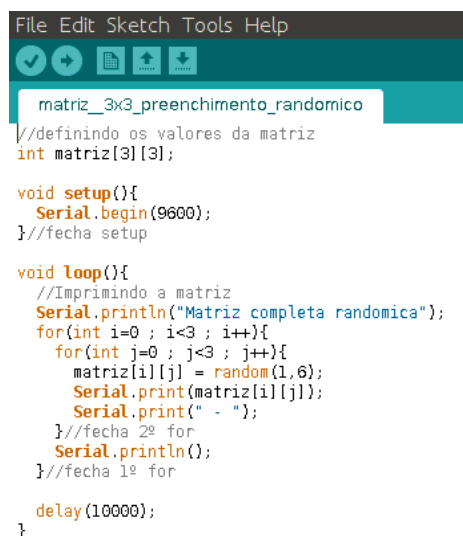
```

Fonte: produzido pelo autor, 2014

2.3.8 Matriz com dados randômicos

O próximo projeto utiliza a mesma ideologia do projeto anterior, porém pode ser pensado na realização de cálculos e soluções matemáticas com matrizes gerando dados randômicos através da função “random”. As Figuras 22 e 23 demonstram o código e a saída no terminal serial dos dados de forma randômica.

Figura 22 –matriz randômica



```

File Edit Sketch Tools Help
matriz_3x3_preenchimento_randomico
//definindo os valores da matriz
int matriz[3][3];

void setup(){
  Serial.begin(9600);
} //fecha setup

void loop(){
  //Imprimindo a matriz
  Serial.println("Matriz completa randomica");
  for(int i=0 ; i<3 ; i++){
    for(int j=0 ; j<3 ; j++){
      matriz[i][j] = random(1,6);
      Serial.print(matriz[i][j]);
      Serial.print(" - ");
    } //fecha 2º for
    Serial.println();
  } //fecha 1º for

  delay(10000);
}

```

Fonte: produzido pelo autor, 2014

Figura 23 - resultado



```

Matriz completa randomica
3 - 5 - 4 -
4 - 1 - 3 -
5 - 4 - 4 -

```

Fonte: produzido pelo autor, 2014

Além desses projetos citados acima, outros podem ser criados a partir da criatividade do professor de matemática. O Arduino pode ser combinado com uma infinidade de sensores e possibilidades que em conjunto podem simplesmente transformar uma sala de aula. A seguir, segue a relação de alguns sensores que podem ser relacionados com a matemática.

O motor servo por exemplo é um motor que pode ser posicionado em 180° através de um joystick, sensor de distância para um leme, flap ou simplesmente para rotacionar a articulação de um braço mecânico. O motor pode ser visualizado no anexo IV.

Com o sensor ultrasônico de distância pode ser medida a distância de um ponto ao outro para realizar cálculos matemáticos diversos. O anexo V mostra este sensor.

O joystick trabalha com os eixos x e y, possibilitando o controle de motores, valores, testes a partir das coordenadas quando o usuário interagir com ele. No anexo VI pode ser observada uma foto do joystick.

A tela de LCD 16x2 serve para imprimir valores na tela, porém a mesma trabalha em formato de matriz 16x2, com isso pode ser trabalhado a lógica de matrizes, cálculos, valores, entre outros, vide anexo VII.

3 CONSIDERAÇÕES FINAIS

Esse artigo ilustrou como o Arduino pode ser útil na Educação Escolar seja ela básica ou técnica, onde o equipamento eletrônico pode auxiliar no desenvolvimento cognitivo dos alunos, deixando o aprendizado de maneira mais simples e com maior significado para os mesmos. Um dos maiores desafios dos professores atualmente é a utilização da tecnologia de forma correta com os alunos em sala de aula, fazendo com que os mesmos pensem, reflitam e desenvolvam a capacidade de utilizar o computador para pesquisar, interpretar e criar novos conhecimentos melhorando seu desempenho escolar e tornando o mesmo mais agradável.

REFERÊNCIAS

- ARDUINO. Arduino. <http://arduino.cc/> Acessado em 31 agosto 2014.
- CITILAB. About S4A. Disponível em: <http://s4a.cat/> Acessado em 31 agosto 2014.
- GARDNER, Howard. Inteligências Múltiplas - A teoria na Prática. Editora Artmed, 2007.
- MATHWORKS, Inc. Arduino Support from MATLAB. Disponível em: <http://www.mathworks.com/hardware-support/arduino-matlab.html> Acessado em 27 agosto 2014.
- MCROBERTS, Michael. Arduino Básico. Tradução de Rafael Zanolli. Editora Novatec, 2011.
- MORAN, José Manuel. **A integração das tecnologias na educação**. Disponível em: http://www.eca.usp.br/prof/moran/site/textos/tecnologias_eduacao/integracao.pdf. 2014. Acesso em: 23:59, 12 outubro, 2014.
- PAPERT, S. (1980). Mindstorms - Children, Computers and Powerful Ideas. New York: Basic

Books, Inc..

PIAGET, Jean. Epistemologia genética. Tradução de Álvaro Cabral. 3. Ed. São Paulo: Martins Fontes, 2007.

SCRATCH. Crie histórias, jogos e animações - compartilhe na internet. Disponível em: <http://scratch.mit.edu/> Acessado em 31 agosto 2014.

VALENTE, José Armando. Informática na Educação: instrucionismo x construcionismo. Disponível em: <<http://www.educacaopublica.rj.gov.br/biblioteca/tecnologia/0003.html>> Acessado em 27 agosto 2014.

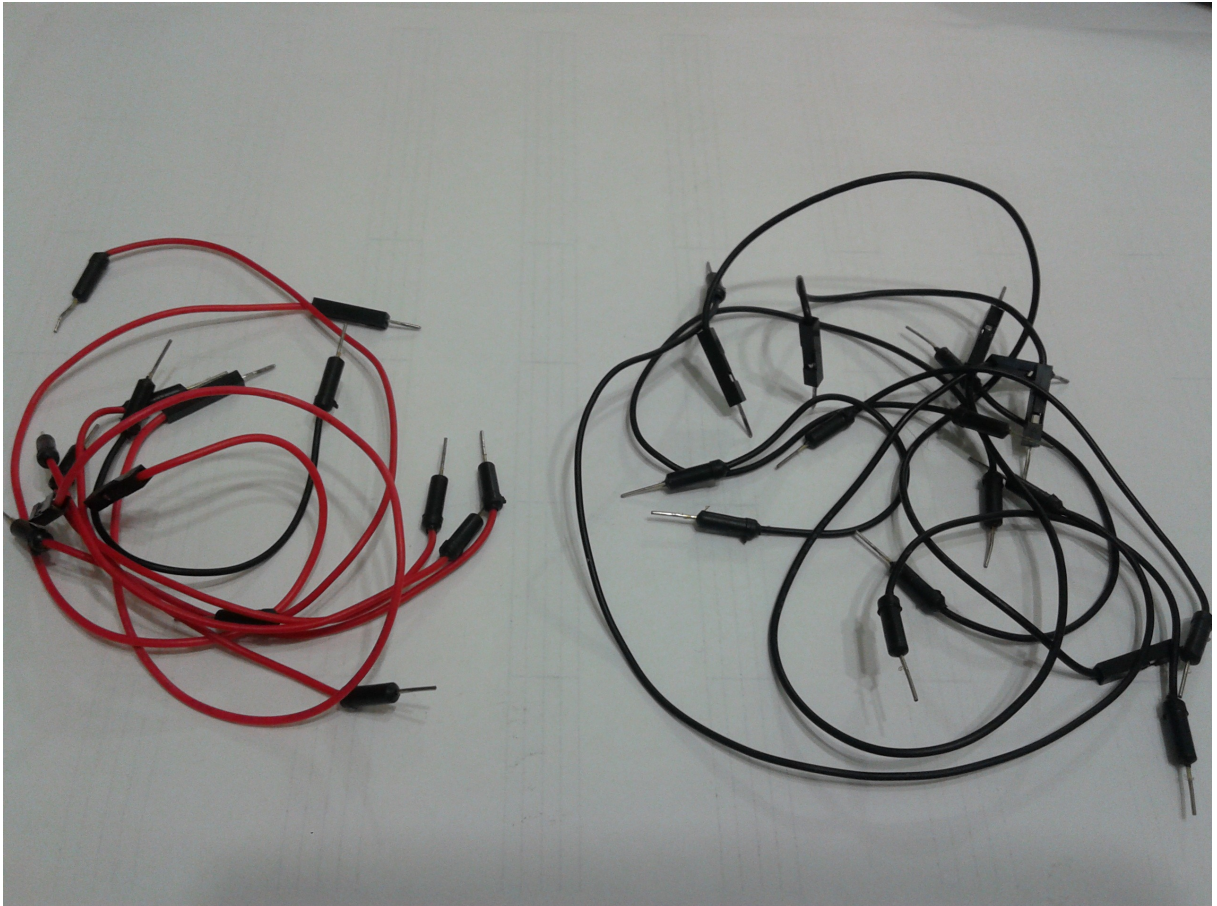
ANEXO I: Imagem dos LED's
Porque a placa eletrônica Arduino deve ser inserida no contexto Escolar das aulas de Matemática

Thiago Espindola Cury
Daniela Rohan Hirschmann



ANEXO II: Imagem dos Jumpers
Porque a placa eletrônica Arduino deve ser inserida no contexto Escolar das aulas de Matemática

Thiago Espindola Cury
Daniela Rohan Hirschmann



ANEXO III: Imagem da breadboard
Porque a placa eletrônica Arduino deve ser inserida no contexto Escolar das aulas de Matemática

Thiago Espindola Cury
Daniela Rohan Hirschmann



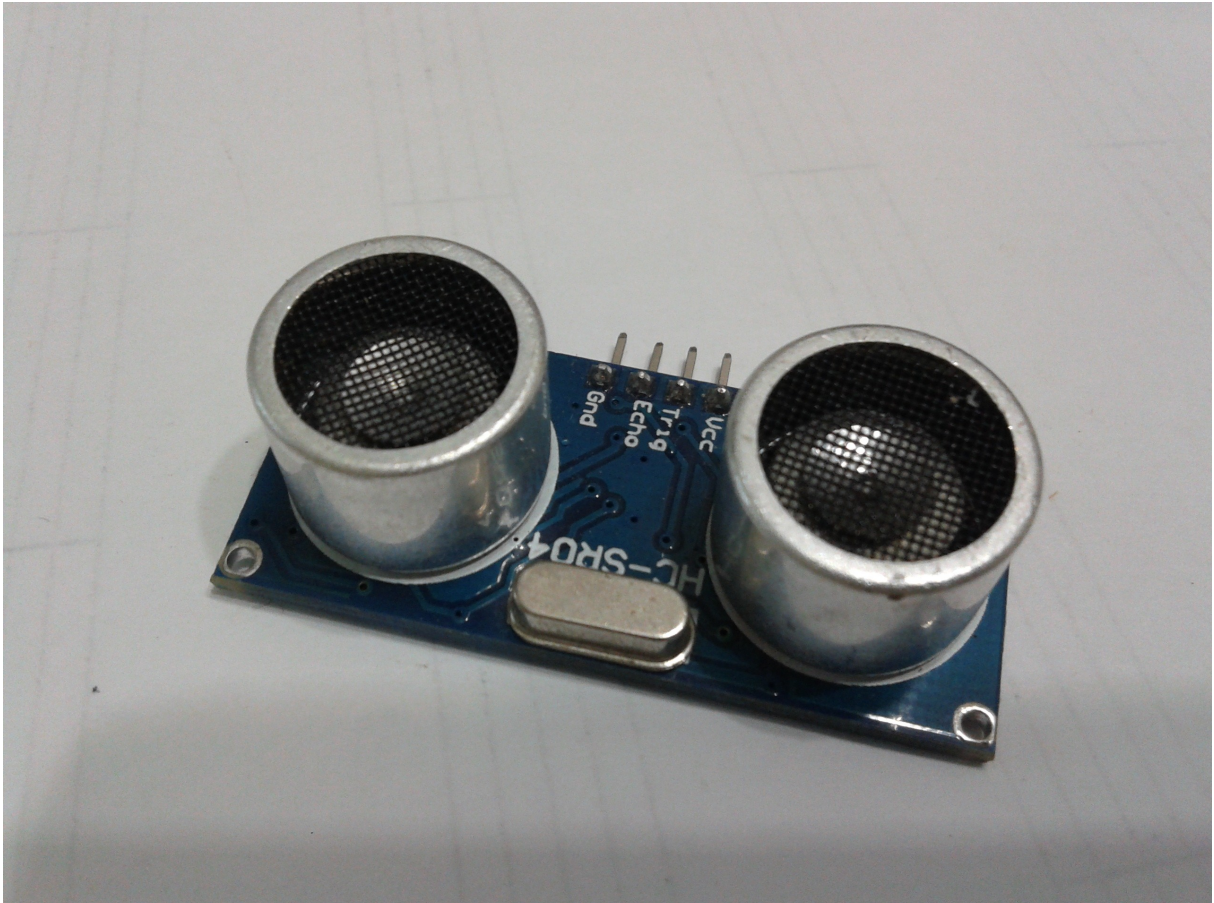
ANEXO IV: Imagem do motor
Porque a placa eletrônica Arduino deve ser inserida no contexto Escolar das aulas de Matemática

Thiago Espindola Cury
Daniela Rohan Hirschmann



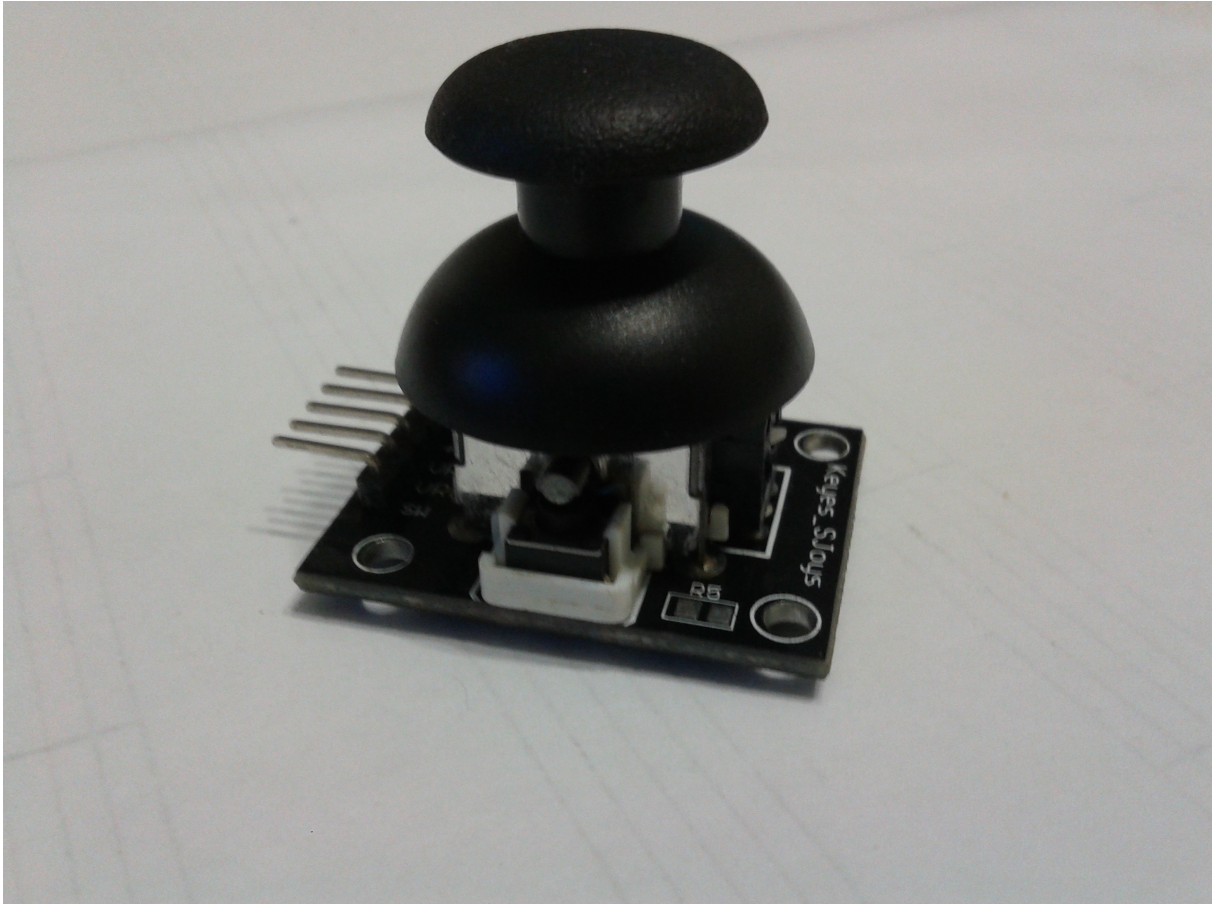
ANEXO V: Imagem do sensor ultrasônico de distância
Porque a placa eletrônica Arduino deve ser inserida no contexto Escolar das aulas de Matemática

Thiago Espindola Cury
Daniela Rohan Hirschmann



ANEXO VI: Imagem do joystick
Porque a placa eletrônica Arduino deve ser inserida no contexto Escolar das aulas de Matemática

Thiago Espindola Cury
Daniela Rohan Hirschmann



ANEXO VII: Imagem do LCD

Porque a placa eletrônica Arduino deve ser inserida no contexto Escolar das aulas de Matemática

Thiago Espindola Cury
Daniela Rohan Hirschmann

