

Arduino Olho Eletrônico

Guilherme Roulim Gotin, Nicolas Pereira Viana, Vinícius Santiago Keller

Análise e Desenvolvimento de Sistemas – Escolas e Faculdades QI

Gravataí – RS – Brasil

{Guilherme} gui.gotin@hotmail.com, {Vinicius} vinicius.keller@hotmail.com,

{Nicolas} nicolasoa@hotmail.com

Resumo. Este artigo fala sobre a história do arduino e sobre o projeto realizado se utilizando do mesmo, ele irá abranger desde a especificação das peças utilizadas em seu desenvolvimento até ideias alternativas de como fazer uso dele.

1. Introdução.

Proposta em Ivrea na Itália o arduino surgiu em 2005, ele veio com intuito de interagir em projetos escolares de forma a ter um orçamento menor que outros sistemas de prototipagem disponíveis na época.

2. História.

O Arduino foi criado em 2005 pelo professor Massimo Banzi na Itália. Banzi queria ensinar para seus alunos conceitos de programação e de eletrônica, porém enfrentava um problema, não havia placas de baixo custo no mercado, e, portanto, isso dificultaria a aquisição do produto por todos os seus alunos. Com isso em mente Banzi decidiu criar uma placa de baixo custo que fosse semelhante a estrutura de um computador para que seus alunos tivessem a oportunidade de aprendizado. A sua placa, nomeada de Arduino, foi um sucesso, recebendo uma menção honrosa na categoria Comunidades Digitais em 2006.

Atualmente existem a venda várias placas semelhantes a original criada por Banzi, porém com componentes construídos por diversas outras pessoas. Isso se deve ao fato que o Arduino original criado por Banzi e seus colegas é Open Source, ou seja, qualquer placa criada com a mesma estrutura do Arduino original e que utilizasse sua linguagem padrão consegue realizar as mesmas funções que um Arduino original.

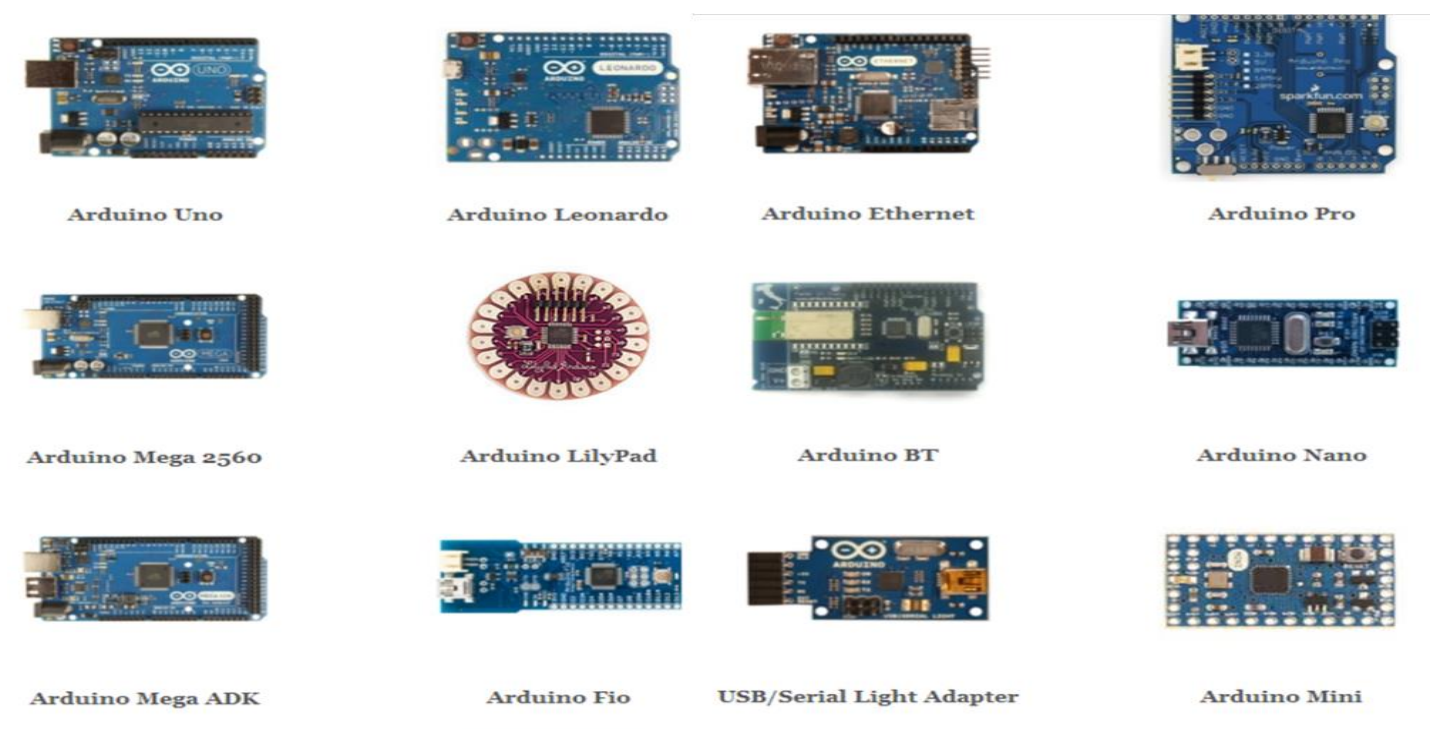


Figure 1. Tipos de Placas de Arduino.

2.1. O Arduino Uno.

A placa Arduino Uno possui dentre seus componentes, 14 pinos digitais de input e output (com 6 deles possibilitando PWM por padrão), 6 inputs analógicos, conexão USB, entre outros. Sua principal diferença entre ele e as outras placas é que o modelo UNO possui o micro controlador Atmega16U2 ao invés do FTDI. Seu nome “UNO” significa 1 (um) em italiano, e significa o lançamento da versão 1.0 do Arduino, e também serve como o modelo de referência para comparação com as versões anteriores do Arduino.

O modelo UNO pode ser alimentado pelo próprio cabo USB, e também por uma fonte de alimentação externa, como por exemplo uma bateria. A placa opera com uma tensão de 6 até 20 volts. O Arduino possui diferentes tipos de pinos de alimentação, são eles: VIN (serve para alimentar o pino com a tensão vinda da fonte externa de alimentação, diferente dos 5 volts que são alimentados pela conexão USB), 5V (alimenta o pino com uma tensão de 5 volts, que pode ser obtida da conexão USB, da alimentação externa ou da tensão recebida pelo pino VIN), 3V3 (uma tensão de 3.3 volts gerada pelo regulador encontrado na placa), GND (é o pino do ground, ou terra), IOREF (dá a tensão de referência na qual o micro controlador opera).

3. O Projeto.

O projeto apresentado, utiliza do arduino para melhorar coisas simples do dia-a-dia, utilizando do sensor de distância ultrassônico para um alerta daquilo que você não está vendo, como estacionar um carro em um lugar apertado, ou tendo uma deficiência visual e não conseguir esquivar-se de objetos que estão na altura de sua cabeça, o sensor ultrassônico basicamente ao chegar em uma distância muito próxima do objeto, vai enviar um alerta para a placa fazendo com que ela ative um sensor, sendo ele um sinal sonoro, ou uma simples lâmpada. Para o sensor de estacionamento usaremos uma placa de arduino simples, apenas com um sensor ultrassônico e com uma lâmpada de LED, já para óculos de deficiente visual, usaremos também um sensor ultrassônico, uma placa de arduino NANO e um buzzer ou um vibracall para chamar a atenção.

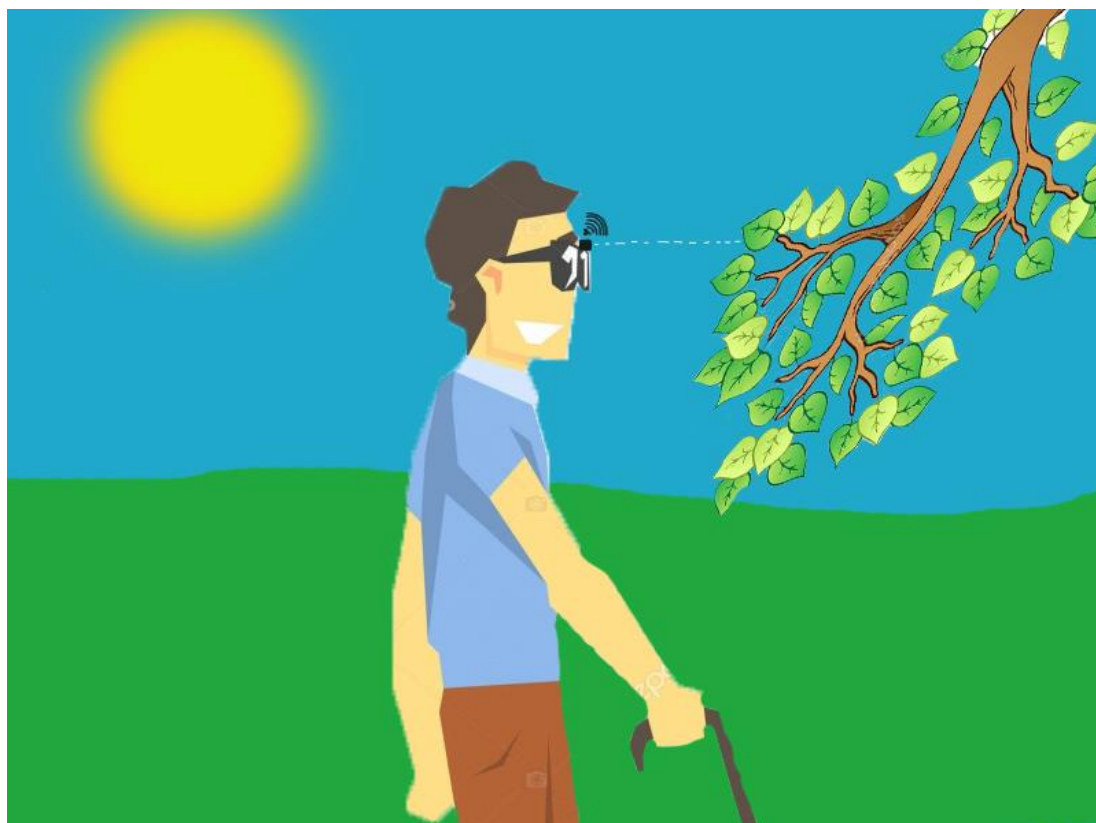


Figure 2. Simulação com buzzer.

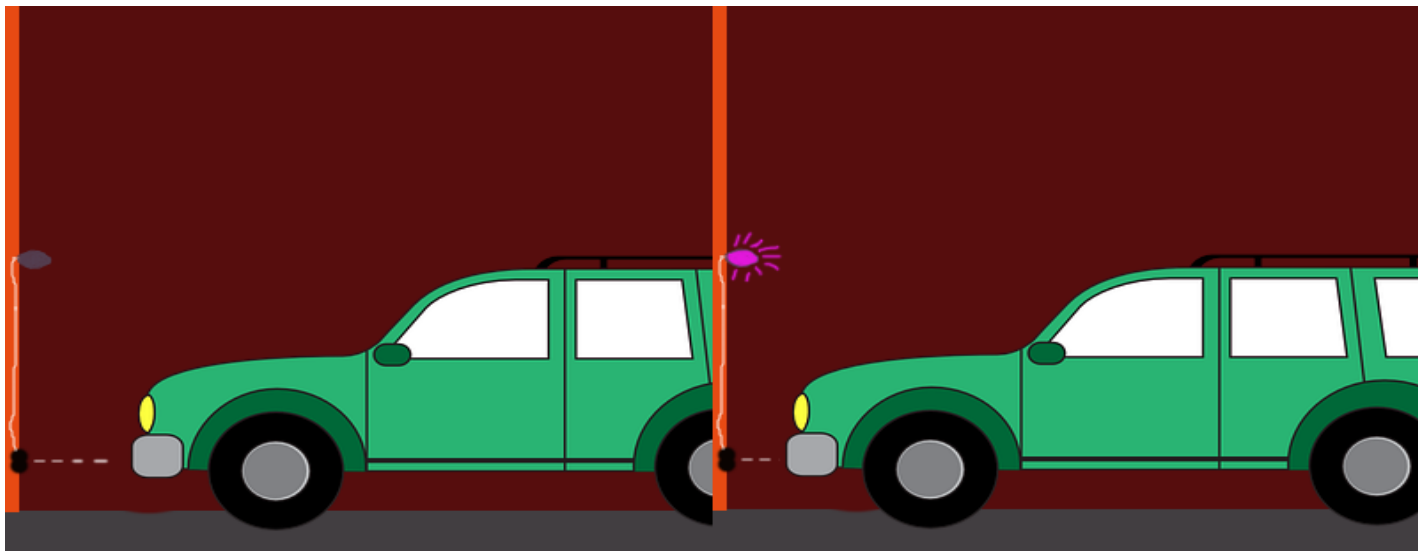


Figure 3. Simulação com LED.

4. Componentes utilizados.

Serão usados poucos componentes para desenvolver o projeto, fazendo com que tenha uma maior economia de energia, e um baixo custo na sua produção.

4.1. Arduino UNO-R3.

A placa de Arduino UNO-R3 é uma das mais simples e mais utilizadas, pelo seu fácil manuseio baixo custo e alta quantidade no mercado, ele contém um botão de reiniciar, uma entrada USB, uma entrada de bateria, quatorze pinos para entrada digital, três pinos negativos, dois pinos 5V, um pino 3,5V e seis pinos Analógicos.



Figure 4. Placa de arduino UNO-R3.

4.2. Arduino NANO V3.

O arduino NANO V3 é uma peça de manuseio um pouco mais complicado que o clássico UNO-R3 contém basicamente a mesma quantidade de pinos digitais de energia e analógicos do arduino UNO-R3, porém com um diferencial de um pino para entrada de energia de 7 a 12 volts.

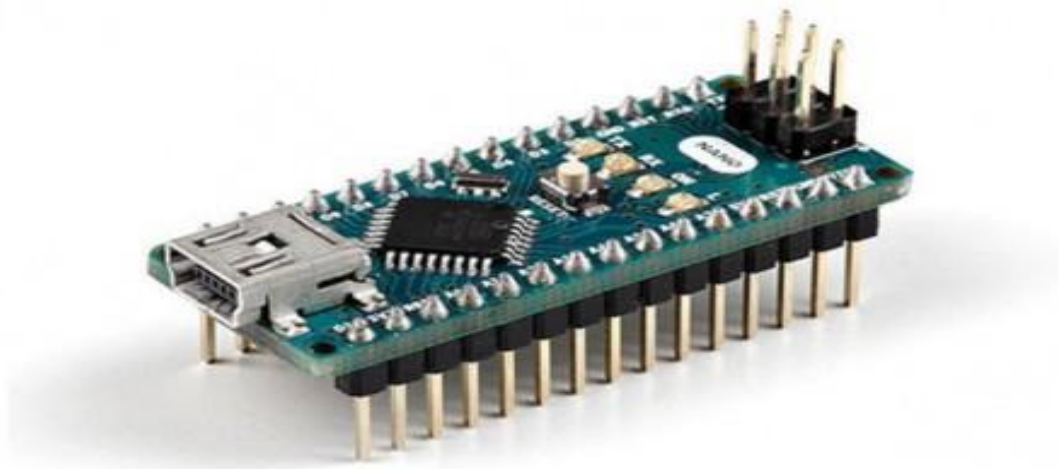


Figure 5. Placa de arduino NANO V3.

4.3. Buzzer ativo.

O buzzer ativo é um componente cuja função é a de um alto falante, tem dois pinos, um negativo e o outro positivo.



Figure 6. Buzzer.

4.4. Motor vibracall.

O motor vibracall de 1.5 a 3 volts, como seu próprio nome já diz tem a função de vibrar, ou seja, chamar a atenção do usuário de uma forma mais discreta que o buzzer.

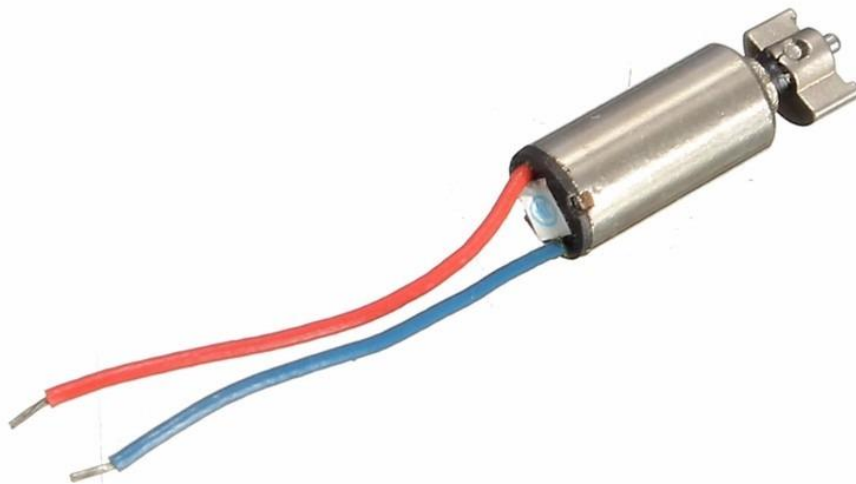


Figure 7. Micromotor Vibracall.

4.5. LED.

O LED tem a função mais simples do projeto, chamar a atenção visualmente.



Figure 8. Led's.

4.6. Sensor ultrassônico HC-SR04.

Este sensor terá a tarefa mais importante dentro do projeto, ele irá com acionar as outras peças do conjunto, mandando um sinal continuo ao se aproximar de mais de um objeto.



Figure 9. Sensor Ultrassônico HC-SR04.

5. Funcionamento.

O funcionamento do HC-SR04 se baseia no envio de sinais ultrassônicos pelo sensor, que aguarda o retorno do sinal, e com base no tempo entre envio e retorno, calcula a distância entre o sensor e o objeto detectado.

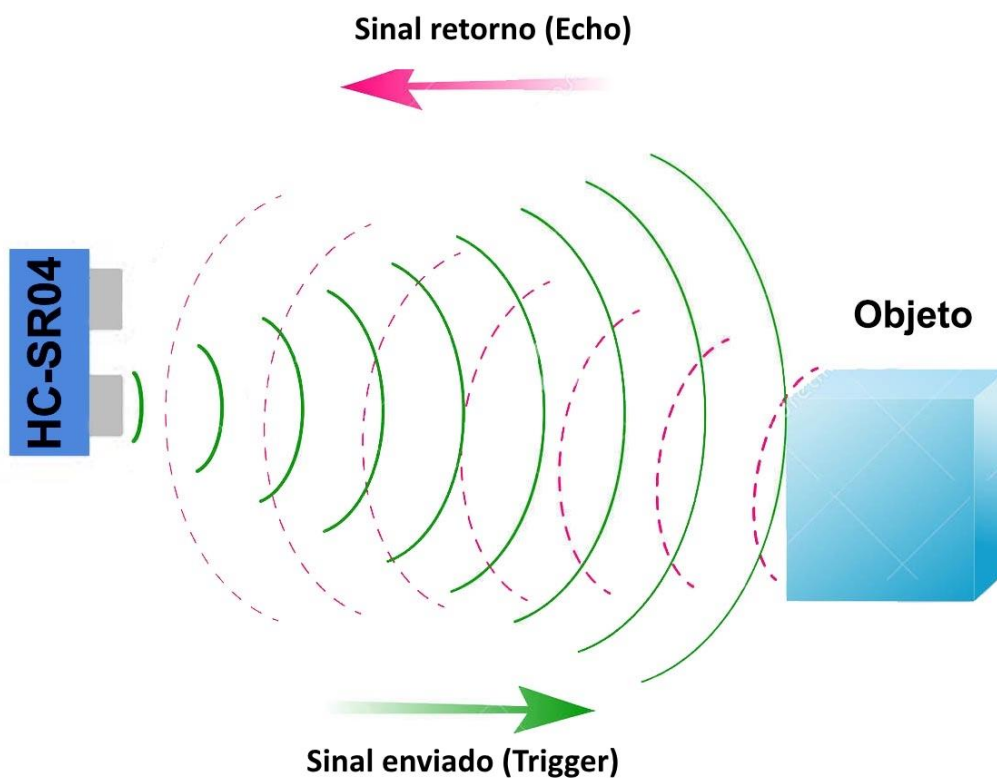


Figure 10. Envio e Retorno de Sinal HC-SR04.

Primeiramente é enviado um pulso de 10µs, indicando o início da transmissão de dados. Depois disso, são enviados 8 pulsos de 40 KHz e o sensor então aguarda o retorno (em nível alto/high), para determinar a distância entre o sensor e o objeto, utilizando a equação $\text{Distância} = (\text{Tempo echo em nível alto} * \text{velocidade do som}) / 2$.

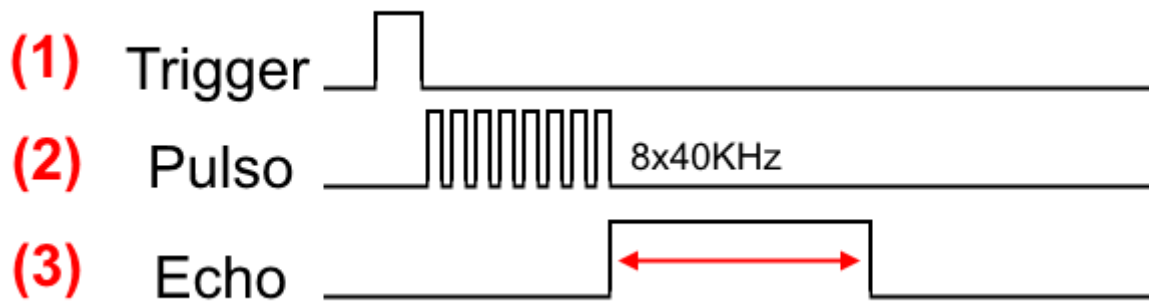


Figure 11. Diagrama de Tempo.

5.1. Conectando sensor ao arduino.

A ligação do sensor ao Arduino vai utilizar, além dos pinos de alimentação, os pinos digitais 4 para o Trigger, e 5 para o Echo.

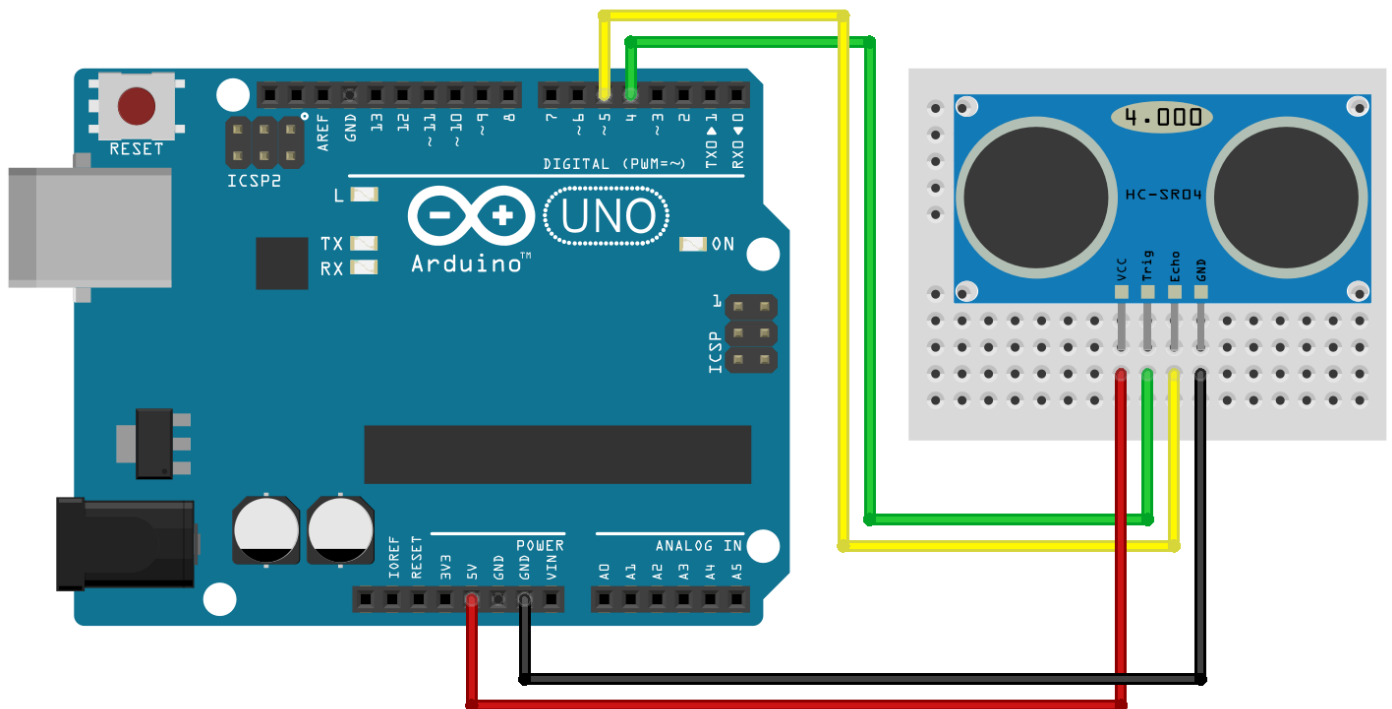


Figure 12. Circuito Arduino e HC-SR04.

6. Programação Arduino.

Antes de tudo, precisa-se entender que um Arduino é um minicomputador que funciona com um software chamado Firmware. Um computador exige uma linguagem especial para funcionar com base em instruções, um Arduino não é diferente e ele usa a linguagem C++, que é uma linguagem muito usada e conhecida, com apenas algumas modificações.

Como todo computador tem sua linguagem para ser programado, ele exige um ambiente de desenvolvimento, ou mais conhecido como IDE (Integrated Development Environment), o Arduino tem uma própria que é onde se passa todas instruções necessárias para que ele faça tudo o que for pedido e, mais importante ainda, que contém um compilador integrado que passa da linguagem C++ para a linguagem do computador.

Para você dar instruções para o Arduino, você tem que entrar no Arduino IDE e vai abrir uma tela para programar um algoritmo, que nesta IDE se chama Sketch. Para explicar melhor, tem que saber mexer com um Blink, que é uma programação básica de ligar ou desligar um Led.

Contém no Arduino IDE dois locais entre chaves para programar: O Void Setup e o Void Loop, porém os dois são ligados, por exemplo, quando nós compilamos tudo o que programamos, e clicamos em compilar, o programa chama o Setup uma vez antes de chamar repetidamente o Loop até o Arduino for desligado.

Os tipos de dados da IDE do Arduino é praticamente igual ao do C++, senão igual, ou seja, uma variável pode carregar um número, um caractere ou até uma série de caracteres formando uma frase. Conforme foi mostrado por Luís Fernando Chavier, todos os tipos de dados desta IDE são:

boolean: valor verdadeiro (true) ou falso (false)

char: um caractere

byte: um byte, ou sequência de 8 bits

int: número inteiro de 16 bits com sinal (-32768 a 32767)

unsigned int: número inteiro de 16 bits sem sinal (0 a 65535)

long: número inteiro de 32 bits com sinal (-2147483648 a 2147483647)

unsigned long: número inteiro de 32 bits sem sinal (0 a 4294967295)

float: número real de precisão simples (ponto flutuante)

double: número real de precisão dupla (ponto flutuante)

string: sequência de caracteres

void: tipo vazio (não tem tipo)

6.1. Código Utilizado.

```
#include <Ultrasonic.h>
```

```
#define pino_trigger 4
```

```
#define pino_echo 5
```

```
Ultrasonic ultrasonic(pino_trigger, pino_echo);
```

```
const int audio = 8;
```

```
void setup()
```

```
{  
  pinMode(audio,OUTPUT);  
}
```

```
void loop()
```

```
{  
  //Le as informacoes do sensor em cm  
  float distancia;  
  long microsec = ultrasonic.timing();  
  distancia = ultrasonic.convert(microsec, Ultrasonic::CM);  
  digitalWrite(audio,LOW);  
}
```

```
while(distancia < 100){  
digitalWrite(audio,HIGH);  
}  
digitalWrite(audio,LOW);  
}
```

7. Conclusão.

Após a realização deste artigo e do projeto, podemos concluir que arduino é uma linguagem com infinitas possibilidades de criação de projetos e fazer o uso da mesma é uma atividade que pode ser muito prazerosa para seus realizadores pois é uma atividade envolvente e que exige muita criatividade dos participantes. Também podemos concluir que o arduino ainda pode vir a ajudar várias pessoas pois através dele surgem projetos muito interessantes que poderão beneficiar pessoas portadoras de deficiências ou dificuldades.

8. Referências Bibliográficas.

THOMSEN, Adilson. *Como conectar o Sensor Ultrassônico HC-SR04 ao Arduino*. Disponível em: <<https://www.filipeflop.com/blog/sensor-ultrassonico-hc-sr04-ao-arduino/>>. Acesso em: 11 de dezembro de 2017.

História arduino. Disponível em: <<http://www.trabalhosfeitos.com/ensaios/Historia-Arduino/49799529.html>>. Acesso em: 11 de dezembro de 2017.

ROSSI, Pedro. *História do arduino*. Disponível em: <<http://vaiquedacertone.blogspot.com.br/2013/04/historia-do-arduino.html/>>. Acesso em: 11 de dezembro de 2017.

Arduino. Disponível em: <<https://pt.wikipedia.org/wiki/Arduino>>. Acesso em: 11 de dezembro de 2017.

Arduino Products. Disponível em: <<https://www.arduino.cc/en/Main/Products>>. Acesso em: 11 de dezembro de 2017.

CHAVIER, Luís. *Programação para arduino*. Disponível em: <<https://www.circuitar.com.br/tutoriais/programacao-para-arduino-primeiros-passos/>>. Acesso em: 17 de dezembro de 2017.

PINTO, Pedro. *Mundo Arduino: Vamos programar?*. Disponível em: <<https://pplware.sapo.pt/gadgets/hardware/mundo-arduino-vamos-comecar-a-programar/>>. Acesso em: 17 de dezembro de 2017.

Tutorial Arduino – primeiros passos. Disponível em: <<https://pandoralab.com.br/aprenda/tutorial-arduino-primeiros-passos/>>. Acesso em: 17 de dezembro de 2017.