



Função map: Saídas e PWM



Esse experimento propõe que a partir da leitura de um potenciômetro ligado em uma porta analógica, e dos valores obtidos, eles sejam convertidos para o padrão PWM (0~255), e enviados para uma porta digital no caso ligada a um led, o que ajudará na percepção do que ocorre fisicamente no uso do PWM através da intensidade com que o led acenderá/apagará.

Nesse experimento serão utilizados os seguintes componentes: 1 potenciômetro 10Kohms, 1 led, 1 resistor de 330 Ohm, alguns fios.

Definição dos componentes:

Potenciômetro

Potenciômetros são dispositivos elétricos que têm a sua resistência elétrica alterada mecanicamente, são muito usados para controlar/alterar as características de entrada/saída de aparelhos eletrônicos, como volume, balanço, graves, brilho, contraste, cor, tempo de funcionamento(em tv's, dvd's, monitores, relógios, ...)...

São também conhecidos como resistores variáveis, ou ainda, reostatos. Possuem internamente uma trilha resistiva (de níquel-cromo ou de carbono), sobre a qual desliza um cursor, que altera a resistência elétrica entre seu conector central e um dos dois laterais(normalmente são três conectores), não são eletricamente polarizados.

O que é PWM

Pulse Width Modulation - Modulação por Largura de Pulso – veja no tutorial no site do Arduino

<http://www.arduino.cc/en/Tutorial/PWM>

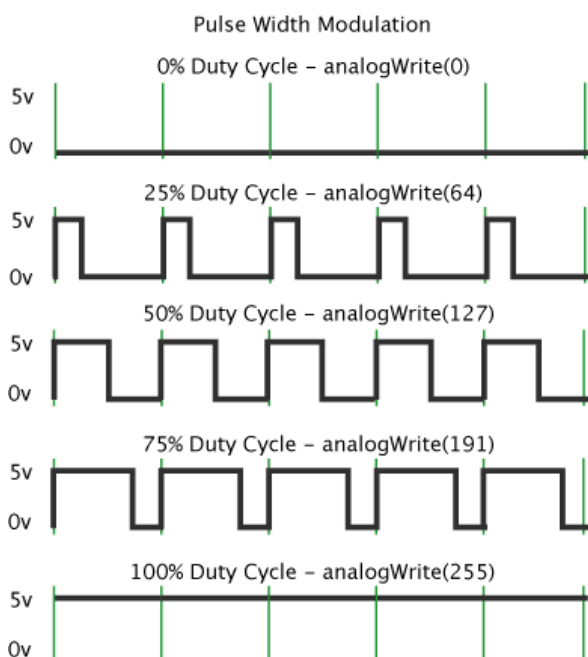
Circuitos digitais só produzem dois números: "0" e "1". Já circuitos analógicos podem ter uma infinidade de variações. Por exemplo, em um circuito digital só podemos ligar ("1") ou desligar ("0") um motor ou uma lâmpada, enquanto que em um circuito analógico podemos controlar em infinitos gradientes o brilho da lâmpada desde o seu estado total de apagamento até o seu brilho máximo. Com um motor acontece o mesmo, podemos controlar em gradientes sua velocidade, desde o seu estado de não rotação até a sua

velocidade

máxima.

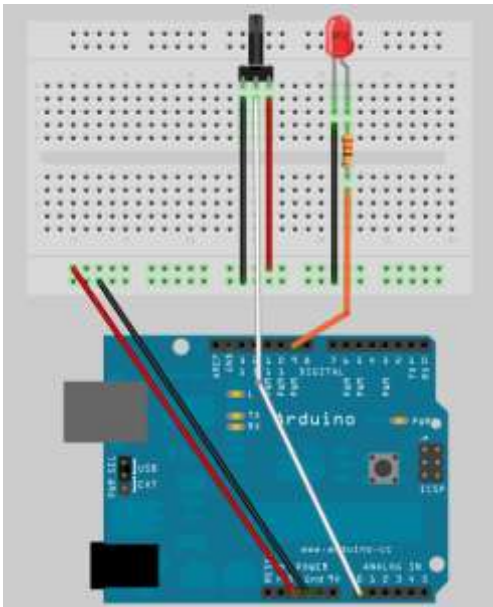
Para um circuito digital poder controlar um circuito analógico - um brilho de uma lâmpada ou a velocidade de um motor - há basicamente duas técnicas. A conversão D/A (Digital/Analógico) e a modulação por largura de pulso (PWM). A conversão digital/analógico usa uma quantidade de bits proporcionais à quantidade de gradientes (brilhos, velocidades, etc) que pretendemos ter. Por exemplo, se forem usados 4 bits, temos uma possibilidade de 16 (2^4) gradientes de brilho/rotação, de 0000 a 1111. O problema desta técnica é que quanto mais gradientes você quiser, mais bits são necessários.

Já a técnica PWM utiliza apenas um bit. Nela é gerada uma forma de onda quadrada onde o ciclo de carga (tempo em que a forma de onda permanece em "1") define a velocidade/brilho do sistema analógico. Por exemplo, supondo uma forma de onda perfeitamente quadrada, onde 50% do tempo ela está em "0" e 50% do tempo ela está em "1", o resultado final será que a lâmpada terá 50% do seu brilho e um motor 50% de sua velocidade. Se configurarmos esta forma de onda para ficar 30% do seu tempo em "1" e 70% do seu tempo em "0", o resultado será um brilho/velocidade de 30% de sua capacidade total. Em resumo, o PWM é uma técnica para conversão digital/analógico usando apenas um bit, onde é gerada uma forma de onda quadrada onde o tempo em que esta forma fica em "1" define o valor da saída (velocidade/brilho).



Obs: o PWM no Arduino é utilizado através dos pinos digitais 3,5,6,9,10,11 e usando a função `analogWrite()`;

Nesse experimento utilizaremos a função `map` que faz uma conversão de escala entre valores, já que o potenciômetro lê valores em um intervalo 0~1023 (10bits) e o `pwm` usa valores no intervalo 0-255 (`8bits`)



Código

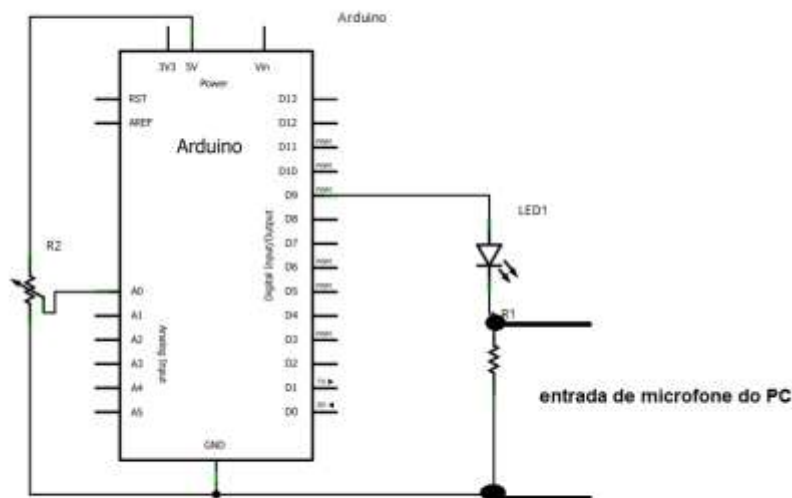
```
int POT; //declara a variavel
int CON;

void setup() {
  pinMode(A0, INPUT);
  pinMode(9, OUTPUT);
}

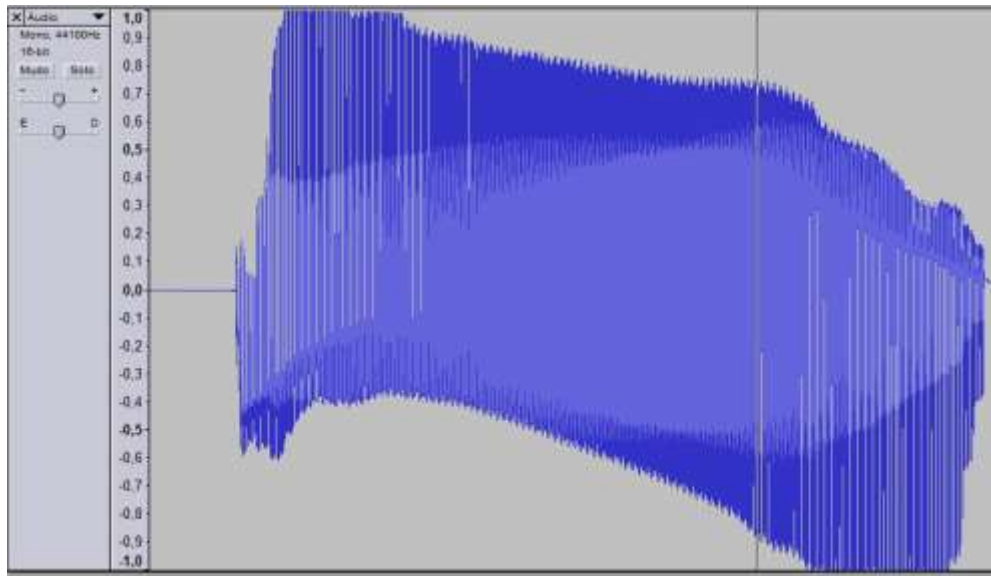
void loop(){
  POT = analogRead(A0);
  CON = map(POT,0,1023,0,255);
  analogWrite(9, CON);
}
```

Observando a saída PWM com a entrada de microfone de um PC

O circuito utilizado segue abaixo



Retirando o sinal através do resistor observa-se utilizando o Audacity. Faça a varredura de uma única vez.

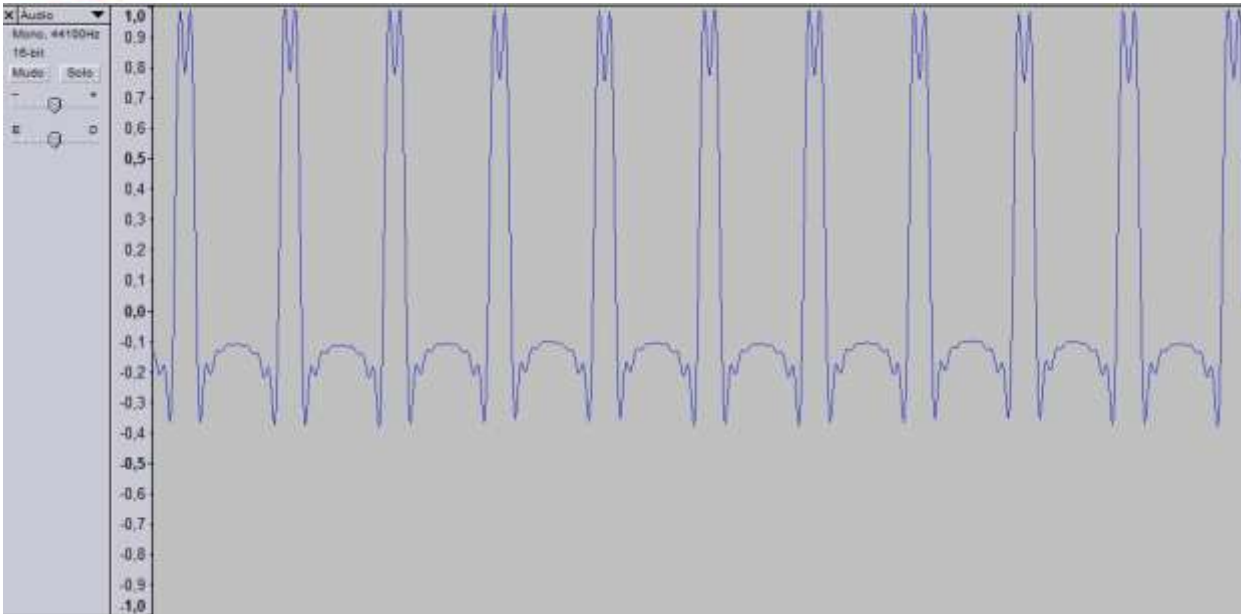


Expandindo este sinal e fazendo a varredura:

Led com baixa intensidade



Led com intensidade intermediária



Led com máxima intensidade



Para baixar o arquivo clique aqui

<https://skydrive.live.com/?cid=59bcf284a2d396aa&sc=documents#!/?cid=59bcf284a2d396aa&sc=documents&nl=1&uc=1&id=59BCF284A2D396AA%21148>

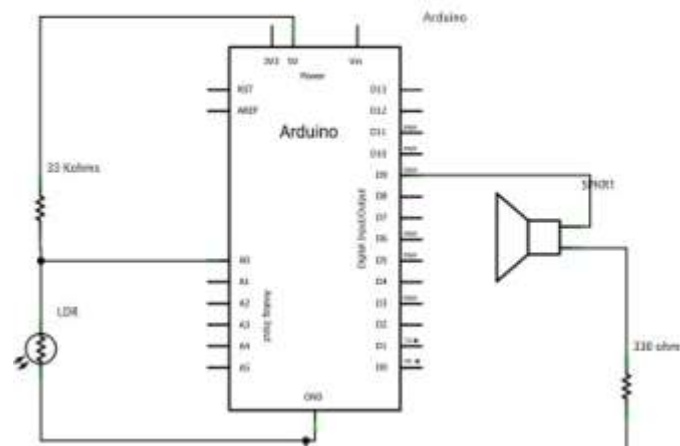
Abra o arquivo no audacity e visualize o sinal a medida que variamos o valor de tensão na porta analógica 0.

Função Map: Sensor de proximidade com sinal sonoro

Substitua o potenciômetro por um LDR e varie a intensidade de luz no LDR. Verifique o que ocorre com a intensidade do LED.

Agora vamos utilizar a uma saída 9 para acionar sinais sonoros. No entanto estes sinais deverão ter frequências dependentes do sinal da porta analógica 0. Assim a função map será associada a frequência.

O esquema do circuito utilizado segue abaixo



Projeto disponível em <http://fritzing.org/projects/sensor-de-proximidade-com-o-arduino-e-sinal-sonoro/>

Código fonte:

```
int LDR; //declara variavel
int som; // variavel som para o alarme

void setup(){
  Serial.begin(9600);
  pinMode(A0, INPUT);// entrada A0
}

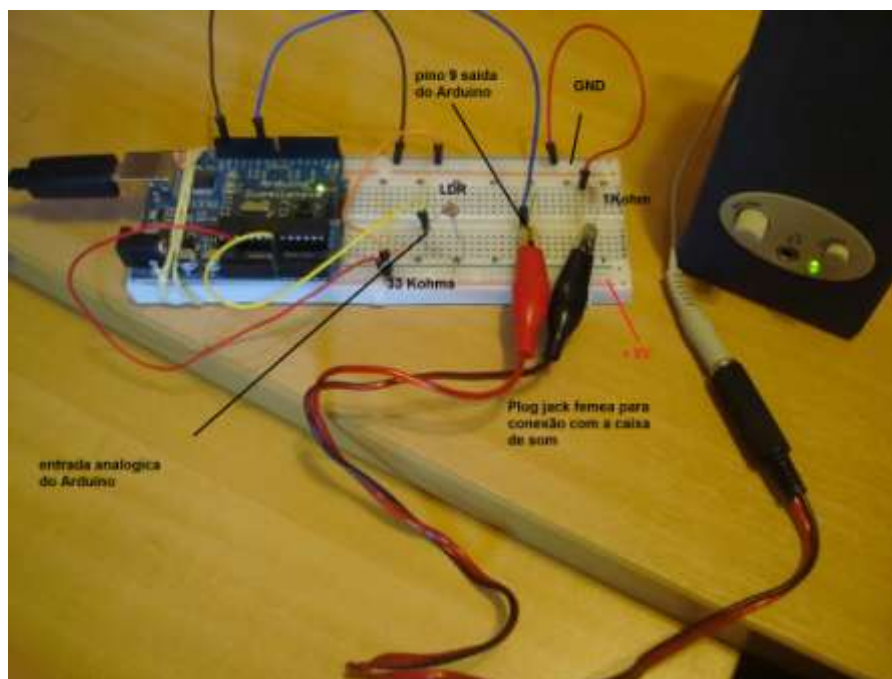
void loop(){
  LDR = analogRead(A0);
  Serial.print(LDR);
  Serial.print(" ");
  Serial.println ( "\t");
  delay(2);

  som = map(LDR, 100, 200, 100, 1000); // toca o som com uma frequencia que
  //varia de 100 a 1000 Hz de acordo com a distancia:
  tone(9, som, 10000);
}
```

Passo a passo:

1. Cole o código acima na IDE do Arduino
2. Imprima os valores lidos no LDR e verifique o intervalo que pretende obter o sinal sonoro
3. A função " som = map(LDR, 100, 200, 100, 1000)" fará esta conversão em valor lido no LDR (aqui está em decimal , varia de 0 a 1023 e está em vermelho) e a saída para o pino 9 será um sinal cuja freqüência esta variando de 100 a 1000 e está em azul)
4. a função " tone(9, som, 10000)" indica; pino de saída , freqüência, que no caso deixaremos dependente da distancia e portanto da leitura no LDR, e o tempo de duração em geral maior do que 100 ms, para retirar o ruído de fundo.

Foto da montagem abaixo



Veja o vídeo deste experimento no link

<http://labduino.blogspot.com/2011/06/som-com-o-arduino-e-um-ldr-e-caixa-de.html>