



OFICINA DE ARDUINO

Prof. M. Sc. Ruben Hernandez
Conceitos e implementações básicas da placa arduino



automática
Semana de Engenharia de Controle e Automação



**UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA**



PROF. M. Sc. RUBEN DARIO HERNÁNDEZ BELEÑO

OFICINA DE ARDUINO

Campinas, 2012

Que você vai aprender neste curso?

- O termo Arduino
- Conhecendo a comunidade Arduino
- Tecnologia Arduino
- Reconhecendo as diferentes placas de Arduino
- Identificando os diversos Shields
- Aplicações específicas com Arduino
- Funcionalidade de cada parte da placa Arduino uno
- Eletrônica
- Voltagem, Corrente e Resistencia
- Lei de Ohm
- Analógico/Digital (Saída/Entrada)
- Programação
- Linguagem de Alto Nível
- Algoritmo
- Sintaxes Arduino
- Definições dos componentes eletrônicos
 - o Resistência
 - o Botão
 - o LED
- Utilidade para a placa Arduino
- Instalando drivers
- Conexão da placa
- Reconhecer a configuração de fabrica
- Verificação da funcionalidade do Arduino
- Funcionalidade da interface
- Sintaxes do programa "Hello World"
- Led intermitente ou ligado
- Entrada/Saída digital
- Retardo
- Flanco de subida/baixada
- Estado do botão
- Console Serial
- Estado do potenciômetro
- Valor analógico
- Variáveis booleanas.
- Saída PWM
- Comparação
- Controle On/off
- Entrada/Saída analógica
- Contadores
- Estruturas e variáveis de programação
- Frequência de intermitência
- Contato magnético
- Uso de funções
- Conceito e controle de PWM
- Estado de um motor
- Saídas PWM "AnalogWrite"





MINI CURSO DA PLATAFORMA ARDUÍNO

PROF. M.Sc. Ruben Dario Hernández B.

Faculdade de Engenharia Mecânica
Universidade Estadual de Campinas



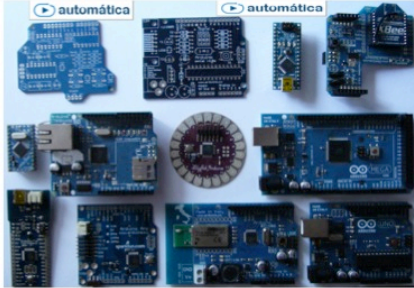
	Introdução	
	Conceitos fundamentais	
	Descrição dos componentes	
	Apresentação do programa arduino	
	Desenvolvimento dos programas	
	Outras Aplicações	
Sumário		

1. **Introdução**
2. **Conceitos fundamentais**
3. **Descrição dos componentes**
4. **Apresentação do programa arduino.**
5. **Desenvolvimento dos programas.**
6. **Outras aplicações**

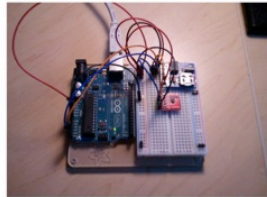
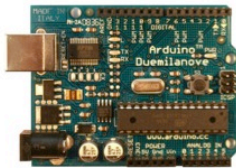
1. **Introdução**
2. Conceitos fundamentais
3. Descrição dos componentes
4. Apresentação do programa arduino.
5. Desenvolvimento dos programas.
6. Outras aplicações

- ☞ Arduino é uma plataforma de software e hardware livre.
- ☞ Foi desenvolvida em 2005 com o objetivo de criar uma plataforma para o desenvolvimento de projetos educativos.
- ☞ Arduino é uma placa baseado no uso do microprocessador ATMEL, sendo bem robustos, imunes ao ruído e as interferências que se apresentem no ambiente.
- ☞ Para sua programação utiliza seu próprio programa, cujo a linguagem é baseada em C com algumas extensões em C++.

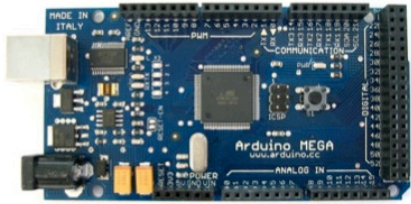
- ☞ É uma solução de baixo custo em comparação a outras placas existentes no mercado.
- ☞ Podem ser desenvolvidos projetos de eletrônica, robótica e automação.
- ☞ Todos seus esquemáticos e componentes eletrônicos se encontram na internet permitindo uma fácil implementação na criação de projetos.
- ☞ Porque é um sistema multiplataforma rodando em Linux, Mac OS e Windows .
- ☞ Pode ser entrelaçado com programas como wiring e processing onde se podem criar aplicações de supervisão.



Há diversos tipos de hardwares e Shields (componentes para auxílio para incrementar o projeto), cada uma delas tem um "codinome" e existem uma variedade que se adaptam de acordo com a necessidade dos projetos.



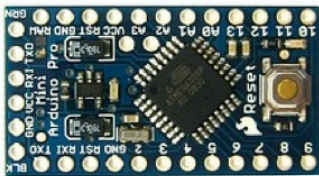
Essa é a última versão da placa USB. Usa um conector USB padrão e contem tudo o que você precisa para programar e usar a placa. É semelhante ao modelo Duemilanove.



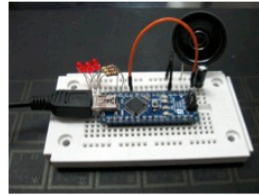
Versão do Mega lançado com o Uno, esta versão apresenta o Atmega2560, que tem o dobro da memória, e usa o 8U2 ATmega para a comunicação serial-USB.



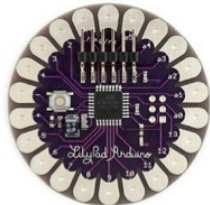
O Arduino BT contém um módulo bluetooth que permite a comunicação sem fio. É compatível com todos os Shields Arduino.



É projetado para usuários avançados que exigem baixos custos, pequenos espaços, mas ainda assim, tem que estar dispostos para fazer algum trabalho extra de montagem.

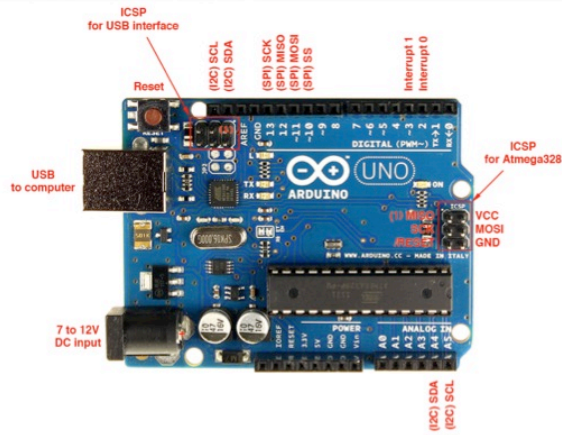


Esta é o menor da placa Arduino. Ela funciona bem em uma protoboard ou para aplicações onde o espaço é pequeno (pequeno mesmo!). Usa um conector Mini USB para se comunicar com o computador.



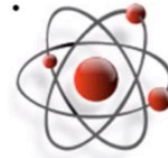
Desenvolvido para aplicações portáteis, é possível "costurar" essa placa em um tecido! Permite fazer coisas curiosas! (camisetas com equalizador? Ah tá...)

Partes do Arduino



1. Introdução
2. **Conceitos fundamentais**
3. Descrição dos componentes
4. Apresentação do programa arduino.
5. Desenvolvimento dos programas.
6. Outras aplicações

O ramo da física e da especialização da engenharia, que estuda e utiliza os sistemas cujo funcionamento baseia-se na conduta e controle do fluxo microscópica de elétrons ou outras partículas eletricamente carregadas.

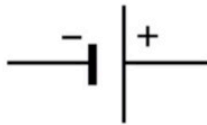


Uma magnitude física que impulsiona os elétrons ao longo de um circuito elétrico condutor fechado, provocando um fluxo de corrente elétrica. Volt (V)



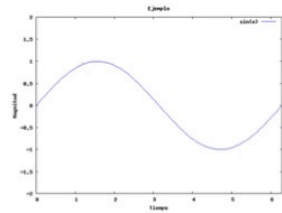
É o fluxo contínuo de elétrons através de um condutor entre dois pontos de potencial elétrico diferente. Na corrente contínua as cargas elétricas sempre circulam na mesma direção, mantendo a polaridade constante.





☞ Trata-se da corrente elétrica na qual a magnitude e direção variam ciclicamente. A forma de onda da corrente alternada mais comumente utilizada é a de uma onda senoidal.

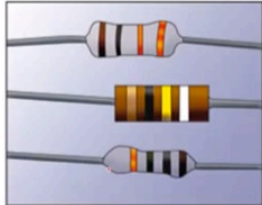
☞ Eletricidade chega a casa e empresas



Trata-se do fluxo de elétrons livres através de um condutor ou semicondutor. A unidade de medida é o Ampère (A).



É a propriedade física que todos os materiais tendem a opor-se ao fluxo de Corrente. A unidade de este parâmetro é o ohm (Ω)



A corrente que flui através de um condutor elétrico é diretamente proporcional à tensão (tensão) e inversamente proporcional à resistência.



$$I = \frac{V}{R}$$

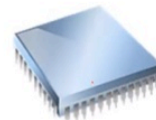
Entradas

Sensores ou transdutores eletrônicos ou mecânicos que medem os sinais (pressão, temperatura, etc.) do mundo físico e convertem em sinais de corrente ou tensão.



Processamento da sinal

O processamento da sinal consiste em componentes eletrônicos ligados entre si para manipular, interpretar e transformar os sinais de tensão e de corrente a partir das entradas.

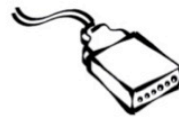


Atuadores, que convertem os sinais de corrente ou tensão em sinais úteis fisicamente.



☞ É a representação de um fenômeno físico, as entradas e saídas de um sistema eletrônico serão sinais variáveis.

☞ Em eletrônica se trabalha com variáveis que assumem a forma de tensão ou corrente.



São aqueles que podem tomar um número infinito de valores entre os limites. A maioria dos fenômenos da vida real são sinais deste tipo. (som, temperatura, voz, vídeo, etc.)



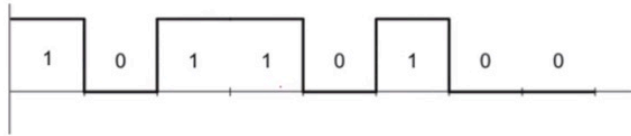
Também chamada de variável discreta, com dois estados distintos, que, portanto, são binários ("0" para "FALSO" e "1" para "VERDADEIRO"). Sendo estas variáveis mais fáceis de se trabalhar.



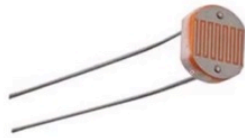
2-5v
Ligado
1
High



0-0.8v
Desligado
0
Low







É uma interface de comunicação de dados digitais, frequentemente utilizados por computadores e periféricos, onde a informação é transmitida byte a byte enviado um único byte de cada vez.



O processo de design, escrever, testar, depurar e manter o código fonte de programas de computador. O código fonte é escrito em uma linguagem de programação. O objetivo da programação é criar programas que apresentem um comportamento desejado.



- ☞ É uma linguagem artificial projetada para expressar processos que podem ser executados por máquinas e computadores.
- ☞ Pode ser usado para criar programas que controlam o comportamento físico e lógico de uma máquina, para expressar algoritmos com precisão.



Tipos de linguagem

- ☞ Linguagem de maquina (1 ou 0).
- ☞ Linguagem assembly
- ☞ Linguagem de alto nível.

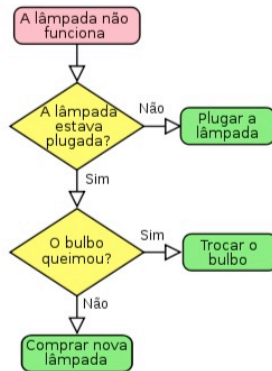


Algoritmo

É um conjunto pré-escrito de instruções ou regras bem definidas, ordenadas e finitos o que permite realizar uma atividade através de etapas sucessivas, sendo este código transparente e legível para o desenvolvedor.



Algoritmo




Linguagem de alto nível

É caracterizada por expressar algoritmos de uma forma apropriada para a capacidade cognitiva humana.

- ☞ JAVA
- ☞ C#
- ☞ Pascal
- ☞ Basic.
- ☞ C++



- ☞ Estruturas
- ☞ Variáveis
- ☞ Operadores matemáticos
- ☞ Operadores de comparação
- ☞ Estrutura de controle
- ☞ Digital
- ☞ Analógico



- ☞ Duas funções principais.
- ☞ Setup()
 - Código de configuração inicial, executa-se só uma vez.
- ☞ Loop()
 - Função que roda logo do setup(), mantém-se executando até desligar o arduino.



Variáveis

É um dado ou um conjunto de dados que muda o seu valor com a execução do programa.

- ☞ Booleano (boolean)
 - True/False
 - Booleano ligado = true;
- ☞ Integer (int)
 - Valor inteiro
 - Int conta = 5;
- ☞ Carácter (char)
 - Armazenar um carácter ASCII
 - Char letra = 'a';
- ☞ Float (float)
 - Usado para armazenar numeros decimais em ponto flutuante
 - float pi = 3.1416;



Operadores matemáticos

Utilizados para a manipulação de números.

- = (igual)
- % (modulo)
- + (soma)
- - (resta)
- * (multiplicação)
- / (divisão)

Operadores usados para comparação logica.

- == (igual a)
- != (diferente a)
- < (menor que)
- > (maior que)

```
If (entrada < 500)
{
    // ação A
} else
{
    // ação B
}
```

```
for (int x=0; x < 100; x++)  
{  
    // ação A  
}
```

```
do  
{  
    x =lerSensor();  
    delay(50);  
} While(x < 100)
```

Processo para a configuração de um pino digital como saída.

- pinMode(pin, OUTPUT);
 - pinMode(a, OUTPUT);
- digitalWrite(pin, value);
 - digitalWrite(a, HIGH);

Processo para a configuração de um pino analógico como entrada.

- int analogRead(pin);
 - int analogRead(c);
- Retorna (0V – 0) (5V - 1024)

Funções

É um bloco de código que tem um nome e um conjunto de sentencias que são executados quando é chamada a função.

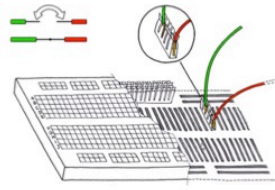
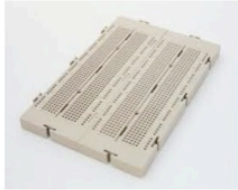
```
- type nomeFunção(parâmetros)
{
    sentencias;
}

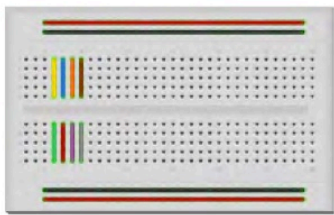
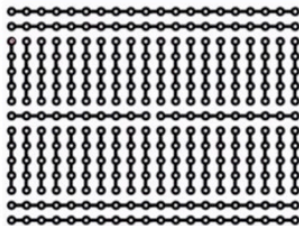
- Int delayVal()
{
    int v;
    v=analogRead(pot);
    v/= 4;
    return v;
}
```

1. Introdução
2. Conceitos fundamentais
- 3. Descrição dos componentes**
4. Apresentação do programa arduino.
5. Desenvolvimento dos programas.
6. Outras aplicações

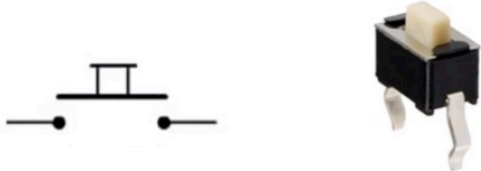
☞ É uma placa reutilizável usada para construir protótipos de circuitos eletrônicos sem solda.

☞ Composta de blocos de plástico perfuradas e numerosas camadas finas de cobre, estanho e fósforo.



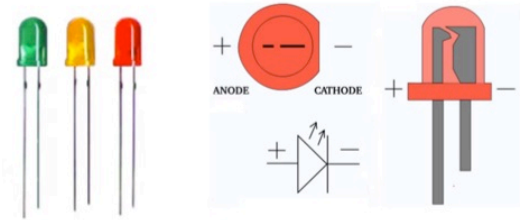


Elemento que permite a passagem ou interrupção da corrente, enquanto é ligado. Quando já não atua, retorna à sua posição de repouso.



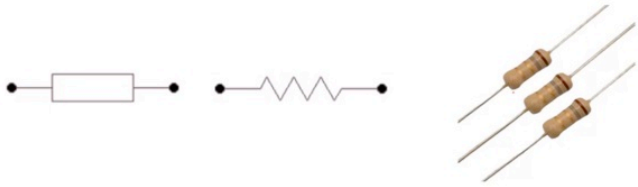
☞ Diodo emissor de luz.

☞ Dispositivo que emite luz quando é ligado corretamente, permitindo o passo da corrente elétrica.



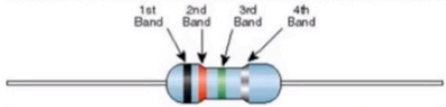
Resistência

- ☞ Opõe-se ao fluxo de corrente.
- ☞ Película de carbono.
- ☞ Tabela de cores
- ☞ Unidade de medida é ohms (Ω)




Resistência

Standard EIA Color Code Table 4 Band: $\pm 2\%$, $\pm 5\%$, and $\pm 10\%$

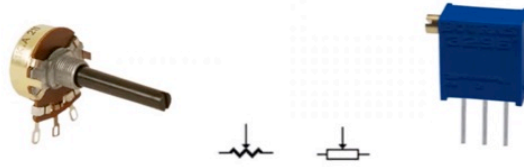


Color	1st Band (1st figure)	2nd Band (2nd figure)	3rd Band (multiplier)	4th Band (tolerance)
Black	0	0	10^0	
Brown	1	1	10^1	
Red	2	2	10^2	$\pm 2\%$
Orange	3	3	10^3	
Yellow	4	4	10^4	
Green	5	5	10^5	
Blue	6	6	10^6	
Violet	7	7	10^7	
Gray	8	8	10^8	
White	9	9	10^9	
Gold			10^{-1}	$\pm 5\%$
Silver			10^{-2}	$\pm 10\%$

Chart Provided By: 

Potenciômetro

- ☞ É um resistor cujo valor da resistência é variável.
- ☞ Unidade de medida é ohms (Ω)




Sensor LDR

- ☞ É uma resistência, cujo valor em ohms varia com as variações da luz incidente.
- ☞ Uma célula fotoelétrica tem um baixo valor de sua resistência à presença de luz, e um valor elevado de resistência na ausência de luz.




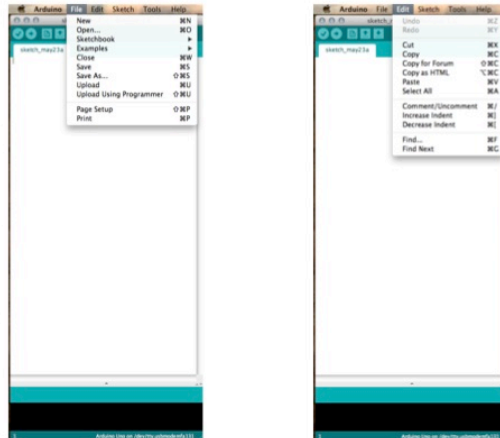
1. Introdução
2. Conceitos fundamentais
3. Descrição dos componentes
4. **Apresentação do programa arduino.**
5. Desenvolvimento dos programas.
6. Outras aplicações





 Introdução
 Conceitos fundamentais
 Descrição dos componentes
Apresentação do programa arduino
 Desenvolvimento dos programas
 Outras Aplicações


Interfase
 Menu file e edit
 Menu sketch e tools
 Menu tools e help

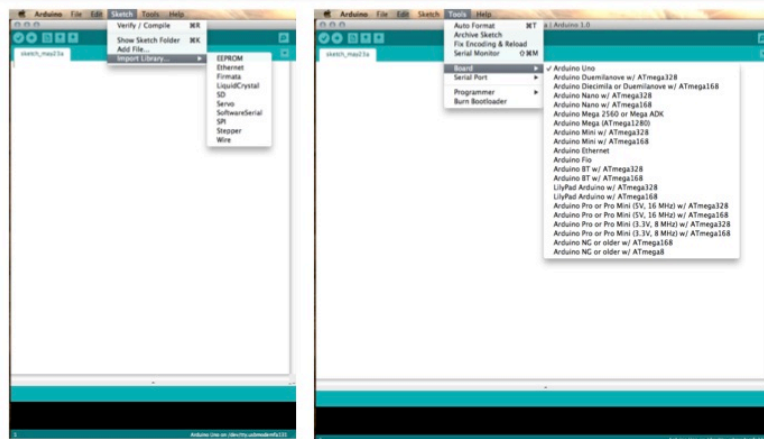


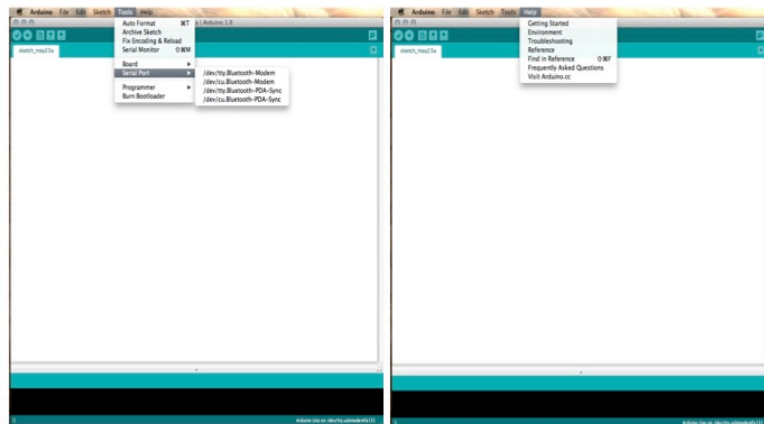



 Introdução
 Conceitos fundamentais
 Descrição dos componentes
Apresentação do programa arduino
 Desenvolvimento dos programas
 Outras Aplicações

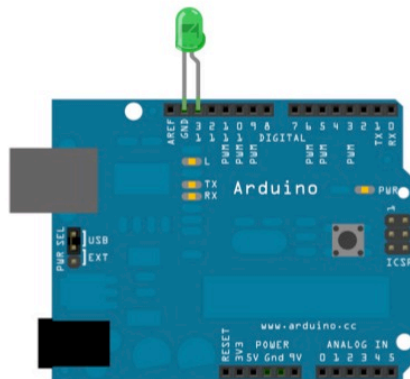
Interfase
 Menu file e edit
Menu sketch e tools
 Menu tools e help







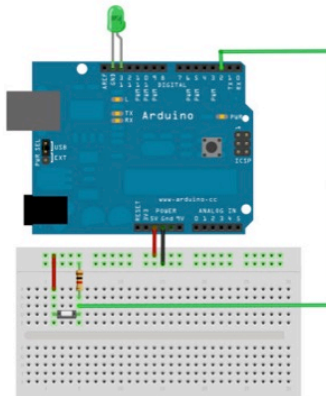
1. Introdução
2. Conceitos fundamentais
3. Descrição dos componentes
4. Apresentação do programa arduino.
- 5. Desenvolvimento dos programas.**
6. Outras aplicações



```

/*
-----
Hello world
-----
Liga e desliga um LED por um segundo
*/
//-----
//Função principal
//-----
void setup()    // Executa-se quando o arduino é ligado
{
  pinMode(13,OUTPUT); // Inicializa o pin 13 como uma saída
}
//-----
//Função repetitiva
//-----
void loop()     // Esta função se executa o instante todo
{
  // quando este ligado o Arduino
  digitalWrite(13,HIGH); // Liga o LED
  delay(1000);           // Temporiza um segundo (1s = 1000ms)
  digitalWrite(13,LOW); // Desliga o LED
  delay(1000);           // Temporiza um segundo (1s = 1000ms)
}

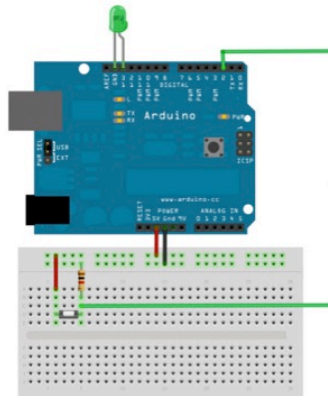
```



```

/*
-----
  Liga um LED com um botão
-----
*/
//-----
//Declara os portos de entradas e saídas
//-----
int botao=2; //Pino onde se encontra o botão, entrada
int led=13;  //Pino onde se encontra o LED, saída
//-----
//Função principal
//-----
void setup()
{
  pinMode(botao, INPUT); //Configura a entrada
  pinMode(led,OUTPUT);   //Configurar a saída
}
//-----
//Função cíclica
//-----
void loop()
{
  //Condicional para saber o estado do botão
  if (digitalRead(botao)==HIGH)
  {
    //Botão oprimido
    digitalWrite(led,HIGH); //Liga o LED
  }
  else
  {
    //Botão NAO oprimido
    digitalWrite(led,LOW); //desliga o LED
  }
}

```

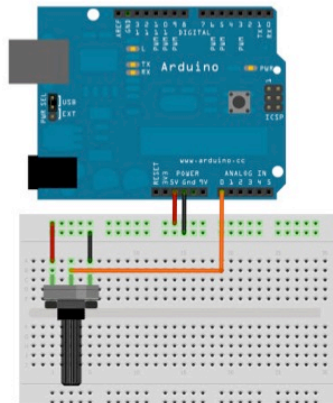


```

/*
-----
  Leitura serial de uma entrada digital
-----
Lê uma entrada digital e apresentar pela tela do computador (console serial) o
estado do botão quando é Oprimido.

*/
//-----
//Declara os pinos de entradas e saídas
//-----
int botao=2;
//-----
//Função principal
//-----
void setup()
{
  //Configuração
  pinMode(botao,INPUT); //Configura o botão como entrada
  Serial.begin(9600); //Inicia a comunicação serial
}
//-----
//Função cíclica
//-----
void loop()
{
  //salva em uma variável inteira o valor do botão 0 ou 1
  int estado = digitalRead(botao);
  //Condicional para saber o estado do botão
  if (estado==1)
  {
    // oprimido
    Serial.println("Ligado"); //Imprime na console serial
  }
  else
  {
    // Não esta oprimido
    Serial.println("desligado");
  }
  delay(100); //Retardo para visualizar os dados na consola
}

```



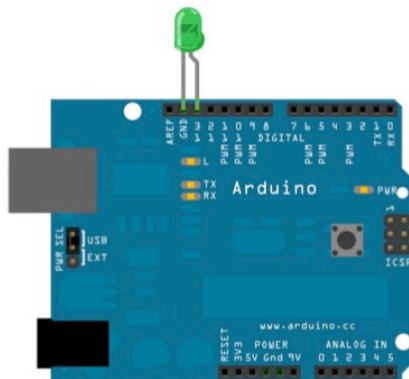
/*

Leitura serial de uma entrada analógica

Lê uma entrada análoga e apresenta pela tela do computador (console serial) o valor logo logo ligar o potenciômetro

*/

```
//-----  
//Função principal  
//-----  
void setup()  
{  
  Serial.begin(9600); //Inicia comunicação serial  
}  
//-----  
//Função repetitiva  
//-----  
void loop()  
{  
  //Salva em uma variável inteira o valor do potenciômetro  
  //0 a 1024  
  int valor= analogRead(A0);  
  //Imprime na console serial o valor da variável  
  Serial.println(valor);  
  //Retardo para a visualização dos dados na console  
  delay(100);  
}
```



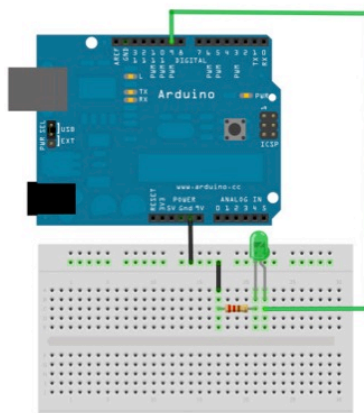
```

/*
-----
  Escrita serial
-----
  É escrever pela tela do computador (console serial) uma letra predefinida, a
  primeira vez que seja escrita ligara um led, se é reescrita pela segunda vez o
  led é desligado

*/

//-----
//Declara os portos de entrada e saídas e variáveis
//-----
int led = 13;      //Pino onde se encontra o LED, saída
char letra;       //Variável onde se guarda a letra
boolean ligado=false; //Estado do LED a primeira vez, desligado
//-----
//Função principal
//-----
void setup()
{
  Serial.begin(9600); //Inicia comunicação serial
  pinMode(led, OUTPUT); //Configura o LED como uma saída
}
//-----
//Função cíclica
//-----
void loop()
{
  //Salva em uma variavel o valor da console serial
  letra=Serial.read();
  // Se é a letra 'L' e além o LED está desligado
  if ( (letra=='L') && (ligado==false) )
  {
    digitalWrite(led,HIGH); // liga o LED
    ligado=true;           // Atualiza o estado do LED
  }
  // Se é a letra 'L' e além o LED está desligado
  else if ( (letra=='L') && (ligado==true) )
  {
    digitalWrite(led,LOW); // desliga o LED
    ligado=false;         // Atualiza o estado do LED
  }
}

```

```

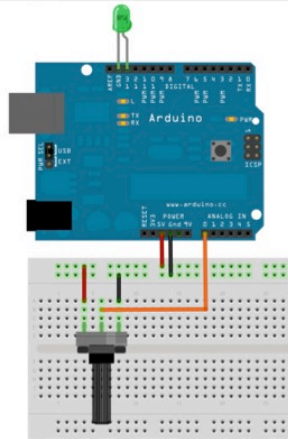
/*
-----
  Liga/desliga um LED de forma proporcional
-----
Programa que liga proporcionalmente um LED quando chega a seu máximo ponto de
intensidade e começa a desligar proporcionalmente.

*/

//-----
//Declara os portos de entradas, saídas e variáveis
//-----
int brilho = 0;          //Variável de brilho inicia em 0
int variacao = 5;       //Variável de incremento configurada de 5 em 5
int led = 9;            //Pino onde se encontra o LED, saída
//-----
//Função principal
//-----
void setup ()
{
  pinMode(led, OUTPUT); //Configura o LED como uma saída
}
//-----
//Função cíclica
//-----
void loop ()
{
  // Escrita analógica (PWM) no LED escrevo o valor de brilho
  analogWrite(led, brilho);
  // Incremento a variável brilho de 5 em 5
  brilho = brilho + variacao;
  // Notação: PWM ----> 0 - 255
  // Se o brilho é 0 o 255
  if (brilho == 0 || brilho == 255)
    variacao = -variacao; //a variação se volta negativa

  delay (30); //Tempo de incremento no brilho
}

```



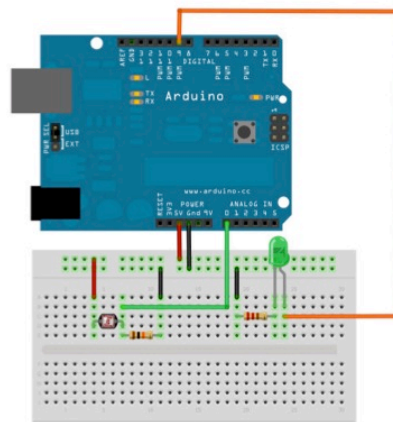
```

/*
-----
  Controle ON/OFF com potenciômetro
-----
Programa que liga um LED quando o valor da entrada analógica comandada
pelo potenciômetro esta em determinado valor, quando este valor varia
o LED se desliga, correspondendo a um sistema com controle ON/OFF

*/
//-----
//Função principal
//-----
void setup()
{
  Serial.begin(9600); //Inicia comunicação serial
  pinMode(13,OUTPUT); //Configura o pin 13 como uma saída
}
//-----
//Função cíclica
//-----
void loop()
{
  //Salva em uma variável o valor da leitura analógica
  int valor = analogRead(A0);
  Serial.println(valor); //Imprime o valor pela console
  //Se o valor é maior o igual a 500
  if (valor >= 500)
  {
    digitalWrite(13,HIGH); //Liga o LED no pino 13
  }
  //Se o valor é menor a 500
  else
  {
    digitalWrite(13,LOW); //Desliga o LED no pino 13
  }

  delay(100); //Retardo de 100ms para obter os dados da consola
}

```



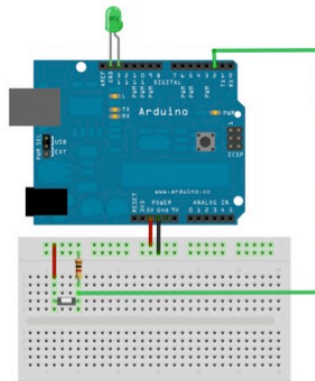
```

/*
-----
  Controle de intensidade de um LED
-----
Programa que liga um LED de maneira proporcional de acordo à quantidade de luz
que incida em uma fotoresistencia LDR.

*/
//-----
//Função principal
//-----
void setup()
{
  Serial.begin(9600);
  pinMode(9,OUTPUT); //Configura o pin 9 como uma saída,PWM
}

//-----
//Função cíclica
//-----
void loop()
{
  //salva em uma variável o valor da leitura analógica da LDR
  int foto = analogRead(A0);
  //Verifica o valor máximo e realiza uma conversão
  int conversao = 780 - foto;
  //Condicional para estabelecer um valor absoluto
  if ( conversao < 0)
    conversao = conversao * -1; //Multiplica por -1 por que é negativo
  //Imprimir dados do valor da leitura analógica da LDR
  Serial.print("LDR : ");
  Serial.print(foto);
  Serial.println("");
  //Imprimir dados do valor da conversão
  Serial.print("Conv : ");
  Serial.print(conversao);
  Serial.println("");
  //Escrita analógica de PWM no LED de acordo à conversão
  analogWrite(9, conversao);
  delay(100); //Retardo para dados na console
}

```

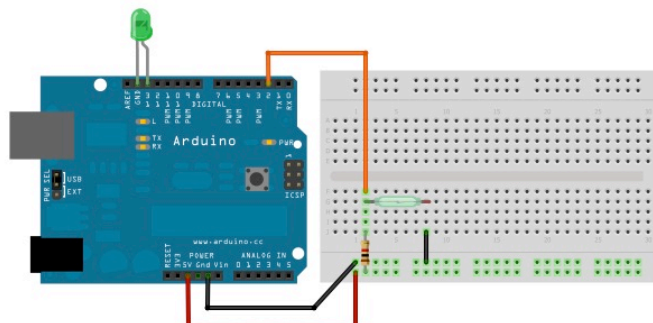


```

/*
-----
  Contador de pulsos
-----
Programa que apresenta pela tela (console serial) o de vezes que o botão tem sido
oprimido, se realiza um processo que de acordo ao número de vezes oprimidas se liga um
LED,

*/
//-----
//Declara os portos de entradas, saídas e variáveis
//-----
int conta = 0; //Variável para salvar o centeio dos pulsos
//-----
//Função principal
//-----
void setup() /
{
  Serial.begin(9600);
  pinMode(2,INPUT); //Configura o pino 2 como uma entrada, botão
  pinMode(13,OUTPUT); //Configura o pino 13 como uma saída, LED
}
//-----
//Função cíclica
//-----
void loop()
{
  // Se o botão esta oprimido
  if ( digitalRead(2) == HIGH )
  {
    // Se o botao nao esta oprimido, flanco de baixada
    if ( digitalRead(2) == LOW )
    {
      conta++; //Acrescenta o contador
      Serial.println(conta); //Imprime o valor pela console
      delay (100); // Retardo
    }
  }
  // Se o valor do contador é 5
  if (conta==5)
  {
    digitalWrite(13,HIGH); //Liga o LED
  }
  // Se o valor do contador é 8
  if (conta==8)
  {
    digitalWrite(13,LOW); // Desliga o LED
  }
}
}

```



```

/*
-----
  Interruptor magnético para gerar uma alerta
-----
Programa que gera uma alarma visual com um LED a partir de um campo magnético
gerado a um reed switch.
*/

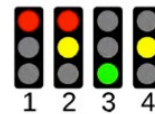
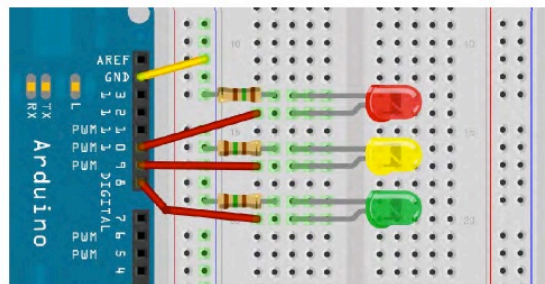
int contato = 2; //Pino assignado ao reed switch
int led= 13;     //Pino assignado ao LED

//-----
//Função principal
//-----
void setup()
{
  pinMode(contato,INPUT); //O reed switch como uma entrada
  pinMode(led, OUTPUT);  //o LED como uma saída
}
//-----
//Função cíclica
//-----
void loop()
{
  // Se o íman se aproxima ao reed switch
  if (digitalRead(contato)==LOW){

    //Ciclo for que vai de 0 a 50, o contador esta
    //na variable a, se repite se é menor de 50
    for(int a=0; a<50; a++){
      digitalWrite(led,HIGH); //Liga o LED
      delay(50); //Tiempo
      digitalWrite(led,LOW); //Desliga o LED
      delay(50); //Tempo
    }
    // Se o íman esta longe do reed switch
  }else{

    digitalWrite(led,LOW); //Mantem desligado o LED
  }
}

```



```

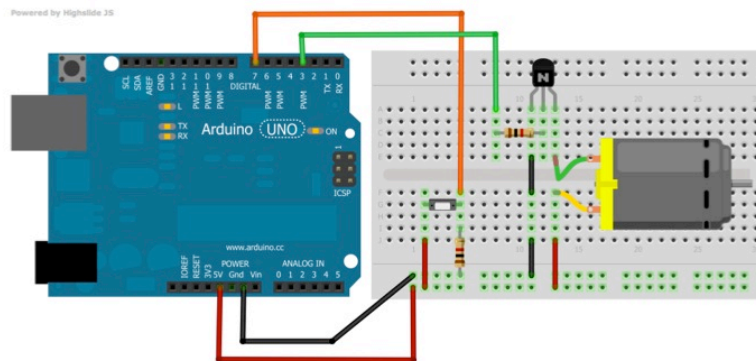
/*
-----
Controle básico de um LED RGB – semaforo
-----

Programa que gera uma alarma visual com um LED a partir de um campo magnético
gerado a um reed switch.

*/
//-----
//Declara portos de entradas, saídas e variáveis
//-----

int ledDelay = 10000; // retardo para o cambio de luz
int vermelhoPino = 10;
int amareloPino = 9;
int verdePino = 8;
//-----
//Função principal
//-----
void setup() {
pinMode(vermelhoPino, OUTPUT);
pinMode(amareloPino, OUTPUT);
pinMode(verdePino, OUTPUT);
}
//-----
//Função cíclica
//-----
void loop() {
digitalWrite(vermelhoPino, HIGH); // liga a luz vermelha
delay(ledDelay); // aguarda 5 seg
digitalWrite(amareloPino, HIGH); // liga a luz amarela
delay(2000); // aguarda 2 seg
digitalWrite(verdePino, HIGH); // liga a luz verde
digitalWrite(vermelhoPino, LOW); // desliga a luz vermelha
digitalWrite(amareloPino, LOW); // desliga a luz amarela
delay(ledDelay); // aguarda o ledDelay
digitalWrite(amareloPino, HIGH); // liga a luz amarela
digitalWrite(verdePino, LOW); // desliga a luz verde
delay(2000); // aguarda 2 seg
digitalWrite(amareloPino, LOW); // desliga a luz amarela
}

```



```

/*
-----
  Controle ON/OFF de um motor
-----
*/

Programa que faz uso de um motor e um botão, Se é mantido oprimido, o motor
deve estar ligado (ON) se não deve estar desligado (OFF)

*/

//-----
//Declara portos de entradas, saídas e variáveis
//-----
int botao=7; //Declara Pino do botão
int motor=3; //Declara Pino do motor
//-----
//Função principal
//-----
void setup()
{
  pinMode(botao,INPUT);
  pinMode(motor,OUTPUT);
}
//-----
//Função cíclica
//-----
void loop()
{
  // Se o botao se encuentra oprimido
  if(digitalRead(botao) == HIGH){

    digitalWrite(motor,HIGH); //liga o motor

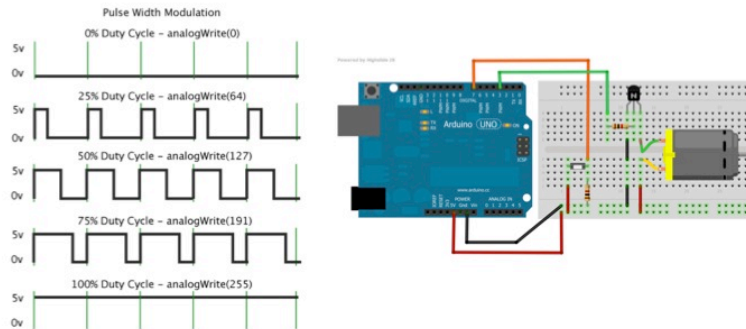
  }else{ //se o botao nao esta oprimido

    digitalWrite(motor,LOW); //desliga o motor

  }

}
}

```



```

/*
-----
  Controle por PWM de um motor
-----
*/

Programa que faz uso de um motor e a Console serial de Arduino, tem a possibilidade de
configurar ao motor em 5 velocidades distintas, desde o teclado do PC pode-se enviar a
velocidade desejada. As 5 velocidades se configuram com 5 PWM distintos.

*/
//-----
//Declara portos de entrada, saídas e variáveis
//-----
int motor=3;    //Declara Pino do motor
//-----
//Função principal
//-----
void setup()
{
  Serial.begin(9600);
}
//-----
//Função cíclica
//-----
void loop()
{
  // Se tem algum valor na consola serial
  if (Serial.available()){
  //Variavel onde se salva o caráter enviado desde o teclado
  char a = Serial.read();
  // Se o caráter ingressado esta entre 0 e 5
  if (a>='0' && a<='5'){
  //Variável para escalar o valor ingressado na faixa de PWM
  int velocidade = map(a,'0','5',0,255);
  //Escrita de PWM ao motor
  analogWrite(motor,velocidade);
  //Mensagem para o usuário
  Serial.print("o motor esta girando à velocidade ");
  Serial.println(a);

  }else{ // Se o caráter ingressado não esta entre 0 e 5
  //Mensagem para o usuário
  Serial.print("Velocidade invalida");
  Serial.println(a);
  }
  }
}
}

```

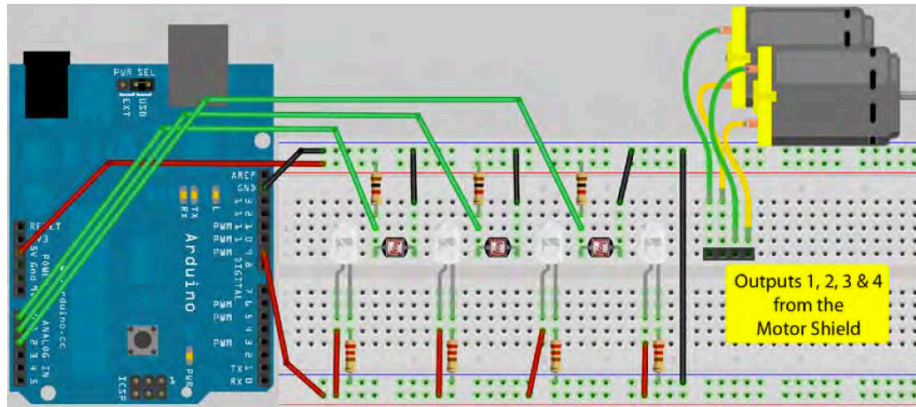

>>Desafio final<<

Rubén Darío Hernández

Oficina de Arduino

1 de julho de 2012

O principal objetivo do desafio é movimentar um veículo em escala por meio da detecção da intensidade de luz, variável medida por três sensores LDR que estão embarcados no veículo, o sensor S1 está situado na direita, o sensor S2 para o frente e o sensor S3 para a esquerda. A montagem eletrônica se apresenta na figura de embaixo, onde baseados nas informações das entradas se atuara com os motores embarcados do veículo.



As informações fornecidas pelos sensores nas entradas analógicas do arduino são, S1=542, S2=504, S3=660. Use o fator de correção entre os valores de 460 a 600 para que as sinais estejam na faixa de 0 a 20 envie pela console serial estas informações e a direção à qual o veículo se desloca, utilize uma velocidade de transmissão de 9600. Implemente seu código de programação tendo em conta a seguinte estrutura que utiliza a função para movimentar os motores do carro, utilize dita função em seu desenvolvimento.

```

#include <Servo.h>
Servo myservo1;
Servo myservo2;
int correcao=460;
int opcao=0;
  
```

```

void setup()
{
  myservo1.attach(9);
  myservo2.attach(11);
  Serial.begin(9600);
}
void loop()
{
  int foto1=analogRead(A0);
  int foto2=analogRead(A1);
  int foto3=analogRead(A2);
  foto1 = foto1-correcao;
  foto2 = foto2-correcao;
  foto3 = foto3-correcao-270;
  Serial.print(foto1);
  Serial.print("----");
  Serial.print(foto2);
  Serial.print("----");
  Serial.println(foto3);
  if(foto1>=380&&foto1<=450)
  {
    opcao=1;
    movimentos(opcao);
    Serial.println("direita");
  }
  if(foto2>=380&&foto2<=450)
  {
    opcao=2;
    movimentos(opcao);
    Serial.println("frente");
  }
  if(foto3>=180&&foto3<=230)
  {
    opcao=3;
    movimentos(opcao);
    Serial.println("esquerda");
  }
}
int movimentos(int mov)
{
  if(mov==1) //Movimenta o carro para a direita
  {
    myservo1.write(140);
    myservo2.write(85);
  }else if(mov==2) //Movimenta o carro para o frente
  {
    myservo1.write(100);
    myservo2.write(85);
  }else if(mov==3) //Movimenta o carro para a esquerda
  {
    myservo1.write(40);
    myservo2.write(85);
  }else //Deixa o carro pairado
  {
    myservo1.write(100);
    myservo2.write(92);
  }
}

```

1. Introdução.
2. Conceitos fundamentais.
3. Descrição dos componentes.
4. Apresentação do programa arduino.
5. Desenvolvimento dos programas.
6. **Outras aplicações.**

- ☞ Robótica móvel
- ☞ Robótica industrial.
- ☞ Automação
- ☞ Domótica



Referências

- ☞ <http://en.wikipedia.org/wiki/Arduino>
- ☞ www.arduino.cc
- ☞ J. Oxel. 2009. "Practical Arduino, cool projects for open source hardware". Technology in action.
- ☞ M. McRoberts. 2010. "Beginning Arduino". Technology in Action.
- ☞ R. E. Herrador. 2009. "Guia de usuario de Arduino". Univesidad de Cordoba.



>>Fim<<
