Robótica com Arduino

Nesta apresentação veremos conceitos sobre contadores, estrutura de condição if, e trabalharemos com a comunicação Serial entre o Arduino e o Computador.

Professor: Paulo Marcos Trentin
Escola CDI de Videira

Problema envolvendo contador

Vamos supor que desejamos fazer um LED piscar por 10 vezes e não mais.

O que precisa ser alterado no código fonte abaixo?

```
// Cria variável global contendo o tempo de atraso
int led = 3;

void setup() {
	// define o pino 2 como sendo de saida
	pinMode(led, OUTPUT);
}
void loop() {
	// Liga o LED
	digitalWrite(led, HIGH);
	// Aguarda 200 milisegundos
	delay(200);
	// Desliga o LED
	digitalWrite(led, LOW);
	// Aguarda 200 milisegundos
	delay(200);
}
```

Conceito do Contador

Nosso cérebro faz isso automaticamente, porém o computador precisa ser programado para fazer uma contagem.

Para permitir o Arduino contar, basta criarmos uma variável que armazene valores inteiros:

```
int contador = 0;
```

Já estamos somando!

Uma vez criada a variável, precisamos incrementá-la a cada vez que o LED piscar.

```
int led= 3;
int contador = 0;
...
void loop() {
        // Liga o LED
        digitalWrite(led, HIGH);
        // Aguarda 200 milisegundos
        delay(200);
        // Desliga o LED
        digitalWrite(led, LOW);
        // Aguarda 200 milisegundos
        delay(200);
        // Incrementa contador
        contador = contador + 1;
}
```

Outras formas de somar ou subtrair

Podemos somar da forma tradicional:

```
contador = contador + 1;
```

Incremento rápido:

```
contador++ ;
```

Neste modo, a variável contador recebe mais 1. Poderíamos decrementar também:

```
contador-- ;
```

Incremento por atalho:

```
contador += 1;
```

A vantagem deste método é que podemos usar um atalho para incrementar de dois em dois e assim por diante:

```
contador += 2;
```

Operador de condição if

Agora, tudo que precisamos é verificar, a cada execução do loop, se o valor da variável contador já chegou à 9 (menor que 10).

```
void loop() {
	// Lê-se: Se o valor da variável contador for menor que, ou igual à 10, então entra dentro do "if"
	if (contador < 10){
		// Liga o LED
		digitalWrite(led, HIGH);
		// Aguarda 200 milisegundos
		delay(200);
		// Desliga o LED
		digitalWrite(led, LOW);
		// Aguarda 200 milisegundos
		delay(200);
		// Incrementa contador. Lê-se: variável contador recebe o valor da variável contador + 1
		contador = contador + 1;
	}
}
```

Operadores de comparação

Muitas vezes precisamos analisar valores, como no exemplo do código anterior. Temos os seguintes operadores (pg 6 da apostila) de comparação:

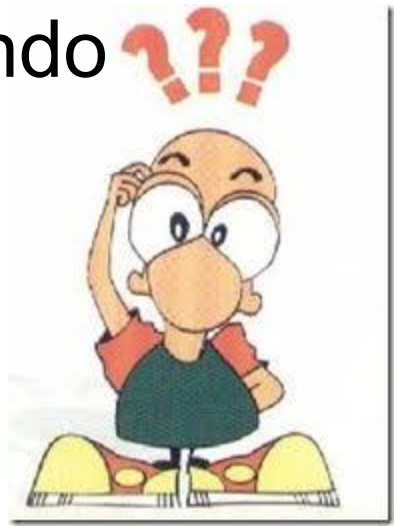
```
==    (igual a)
!= (diferente de)
<     (menor que)
>     (maior que)
<=    (menor que ou igual)
>=    (maior que ou igual)
```

Muito teórico! Não acredito!

Como vou confiar que o contador está mesmo sendo incrementado?

Como ter certeza disso?

Como ver em tempo real o contador sendo incrementado?



Solução: Comunicação Serial!

- Permite comunicações de longas distâncias (o limite é a capaciância dos cabos, 80 metros em alguns casos funciona normalmente)
- Muito estável
- USB é o Universal Serial Bus
- Com Arduino, velocidades de até 115.200 baud são aceitas

Limite de velocidade

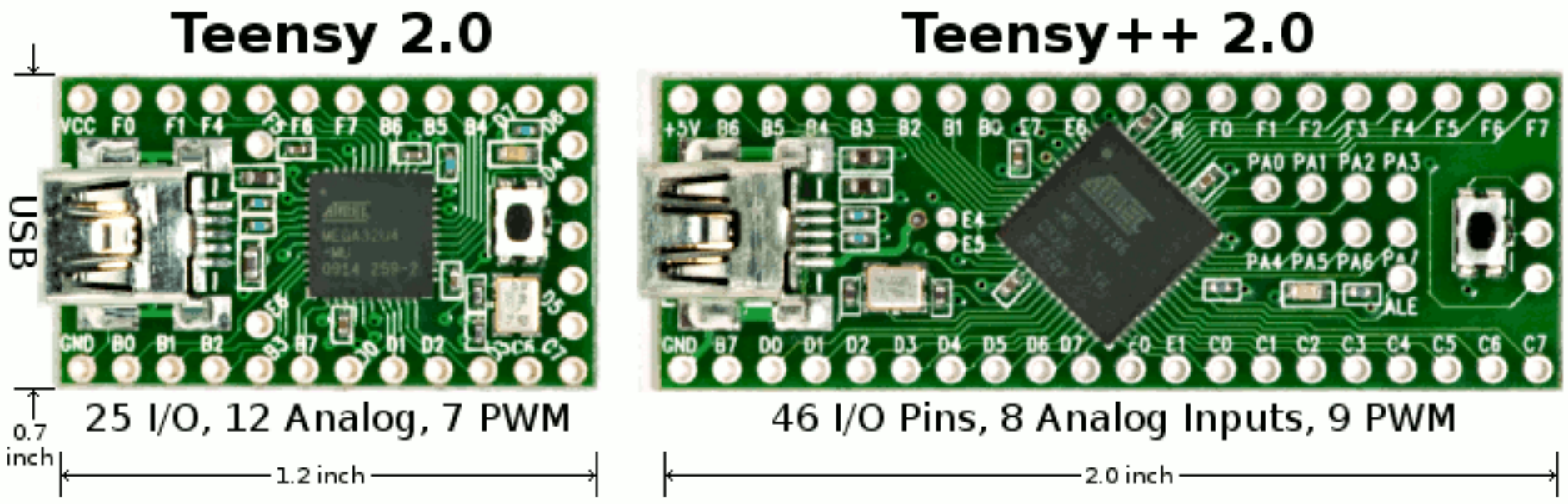
O Arduino Duemilanove utiliza o chip FT232R para comunicar-se com o computador.

Esse chip, segundo datasheet do fabricante ft232r (pg 27), suporta transferências até 1M baud.

O console serial da IDE do Arduino, por usa vez, suporta até 115200 baud por segundo.

E seu precisar mais velocidade?

A solução então é usar um chip que já venha de fábrica com suporte à USB, exemplo disso é o pr[ojeto Teensy]:



Qual é a velocidade do Teensy

A comunicação entre o computador e o Teensy ocorre à 12Mbps.

Para maiores detalhes sobre a diferença de tempo de transmissão do Arduino padrão e o projeto Teensy, acesse este link:

http://www.pjrc.com/teensy/td_serial.html

Como começar a trabalhar com a USB Serial do Arduino?

Veja como é simples enviar dados do Arduino para o computador:

```
void setup() {
    // Inicia comunicação serial a 9600 baud
    Serial.begin(9600);
}
void loop() {
    // Envia dados para a serial
    Serial.println("Executou o loop");
    // Aguarda 1 segundo
    delay(1000);
}
```

Entendendo o Serial.begin(9600);

Serial é um objeto interno do Arduino.

Ele tem um método, chamado begin, ou seja, começar.

Aqui definimos que vamos usar a **Serial** do Arduino, e que a velocidade de transmissão é 9600 baud.

Entendendo o Serial.print();

Serial é um objeto interno do Arduino.

Ele tem um outro método, chamado print, ou seja, imprimir.

Esse método, envia para a serial o conteúdo passado como parâmetro.

Qual a diferença de print() e println() ?

O método print() apenas envia o conteúdo para a serial.

Já println() além de enviar o conteúdo, também imprime uma quebra de linha após o envio.

Mais informações sobre Serial.print() aqui:

http://arduino.cc/en/Serial/Print

Exercício 1

Faça um programa que utilize o método print() e println() do objeto **Serial**

O print deve imprimir a mensagem: "imprimi algo sem quebra de linha"

O println() deve imprimir a mensagem: "imprimi algo com quebra de linha"

Exercício1 - resposta

```
void setup() {
    // Inicia comunicação serial a 9600 baud
    Serial.begin(9600);
}
void loop() {
    // Envia dados para a serial
    Serial.print("imprimi algo sem quebra de linha");
    // Envia dados para a serial
    Serial.println("imprimi algo com quebra de linha");
    // Aguarda 5 segundos
    delay(5000);
}
```

Enviando valores de variáveis

Verifique o valor da variável contador:

```
int contador = 0;
void setup() {
    // Inicia comunicação serial a 9600 baud
    Serial.begin(9600);
    // Envia dados para a serial
    Serial.print("Valor da variavel contador: ");
    Serial.println(contador);
}
void loop() {}
```

Exercício 2 - voltando ao contador

- Crie um novo programa com um contador
- Ao iniciar o programa deve enviar na serial: "Iniciando programa contador..."
- Este contador deve ser incrementado com + 2 a cada loop no intervalo de 1 segundo
- A cada loop, deve ser enviado para a serial o valor atual do contador

Recebendo dados do computador

Podemos facilmente enviar dados do computador para o Arduino

Para isso usaremos o "Serial Monitor" da IDE

Recebendo dados do computador - código fonte

```
char byteRecebido;

void setup() {
      // Inicia comunicação serial a 9600 baud
      Serial.begin(9600);
      Serial.println("Digite algo e pressione Enter");
}
void loop() {
      // Verifica se chegou algo pela serial
      if (Serial.available() > 0){
            // Lê a serial e coloca o valor na variável byteRecebido
            byteRecebido = Serial.read();
            Serial.print("Eu recebi: ");
            // Imprime na Serial o valor recebido
            Serial.println(byteRecebido);
      }
}
```

Entendendo o Serial.available();

Serial é um objeto interno do Arduino.

Ele tem um método, chamado available, ou seja, disponível.

Se houver alguma informação para ser lida na Serial, retorna o número de bytes a serem lidos, do contrário, retorna 0.

Entendendo o Serial.read();

O método read, lê da serial um byte e o retorna para ser usado. Quando usamos:

```
byteRecebido = Serial.read();
```

Estamos dizendo para ler a Serial, e inserir o valor lido na variável byteRecebido

Como verificar se uma letra foi digitada?

```
char byteRecebido;
void setup() {
	// Inicia comunicação serial a 9600 baud
	Serial.begin(9600);
	// Exibe mensagem de boas vindas
	Serial.println("Insira alguma letra e pressione Enter");
}
void loop() {
	// Verifica se chegou algo pela serial
	if (Serial.available() > 0){
		// Lê a serial e coloca o valor na variável byteRecebido
		byteRecebido = Serial.read();
		Serial.print("Eu recebi: ");
		// Imprime na Serial o valor recebido
		Serial.println(byteRecebido);
		// Verifica se a letra a foi digitada
		if (byteRecebido == 'a'){
			Serial.println("Voce digitou a tecla a");
		}
	}
}
```

Exercício 2

Faça um programa que receba do computador, via serial, comandos do usuário

Quando o usuário digitar a tecla L, seu programa deve ligar todos os LEDs da protoboard

Quando o usuário digitar D, então todos os LEDs devem ser desligados