

Aula 02 - SEEL 2019

Minicurso de Arduino

Comunicação Serial, Portas Analógicas, Bibliotecas e Motores

Adriano Rodrigues

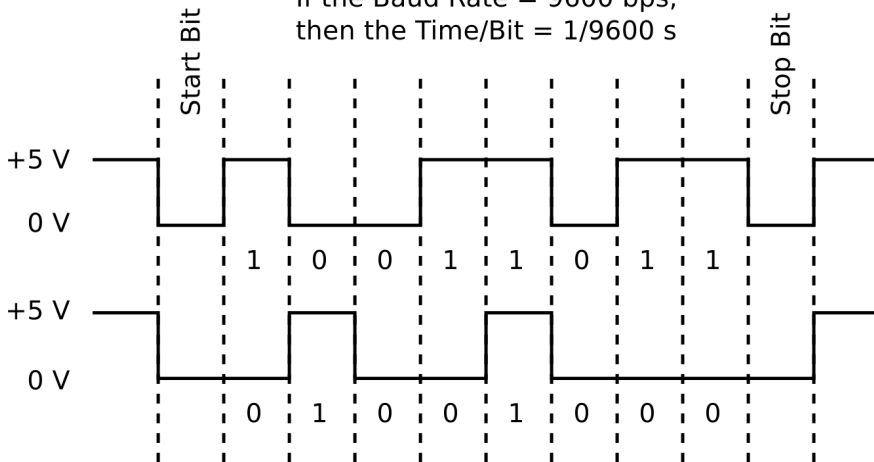
24 de outubro de 2019

- Comunicação Serial;
- Sinais Analógicos:
 - Entradas Analógicas;
 - “Saídas” Analógicas (PWM).
- Motor CC;
- Bibliotecas;
- Servo Motor.

- 1 Conteúdo
- 2 Índice
- 3 Comunicação Serial
- 4 Exercício 01
- 5 Analógico vs Digital
- 6 Funções Analógicas pt1
- 7 Exercício 02
- 8 Potenciômetro
- 9 PWM
- 10 Funções Analógicas pt2
- 11 Exercício 03
- 12 Tarefa 01
- 13 Acionamento dos Motores
- 14 Exercício 04
- 15 Tarefa 02
- 16 Servomotores
- 17 Bibliotecas
- 18 Exercício 05
- 19 Tarefa 03
- 20 Fim!

Comunicação Serial

If the Baud Rate = 9600 bps,
then the Time/Bit = $1/9600$ s



Setup():

- Serial.begin(*baudrate*) *baudrate* = int 300, 600, 1200, 2400, 4800, **9600**, 14400, 19200, 28800, 38400, 56700, 115200.

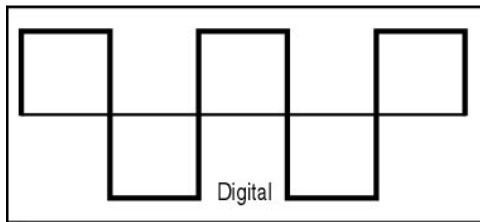
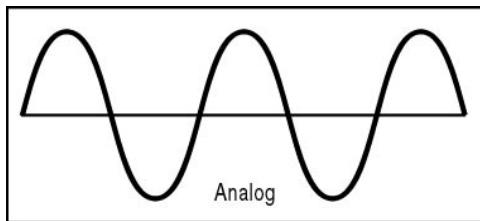
Loop():

- Serial.print(*texto*); → *texto* = string ou char.
- Serial.println(*texto*); → Imprime e pula uma linha.
- Uma string pode ser escrita diretamente entre aspas (“exemplo”) e um caractere entre aspas simples ('a').
- Serial.available(); → Retorna a quantidade de bytes a serem lidos.
- Serial.read(); → Retorna o byte ou caractere lido na pilha.
- Serial.parseInt(); → Retorna número inteiro lido na serial.
- atoi(*string*); → Retorna um número inteiro.
Exemplo *int* val = atoi(“12345”) → val = 12345.

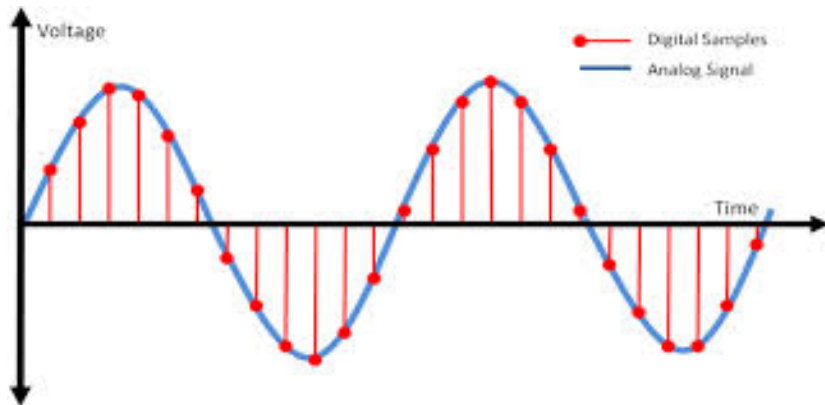
Exercício 01 - Comunicação Serial de Inteiros

```
1 // Comunicacao Serial de Inteiros - Adriano Rodrigues.
2 int buf = 0;
3 void setup() // Sempre utilizar Serial begin no setup.
4 {
5     Serial.begin(9600); // Configura Serial para 9600 de baud rate.
6 }
7 void loop()
8 {
9     if (Serial.available()>0)
10    {
11        buf = Serial.parseInt(); // Interpreta o dado lido como int.
12        Serial.println(buf*2) ; // Imprime BUF multiplicado por 2.
13    }
14 }
```

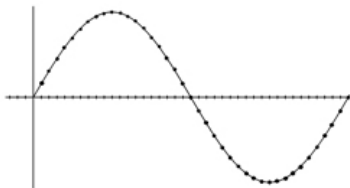
Analógico vs Digital



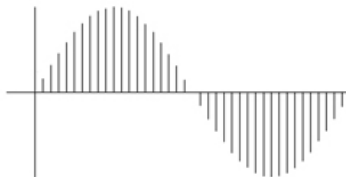
Analógico vs Digital



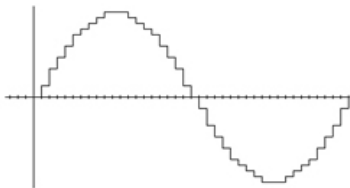
Analógico vs Digital



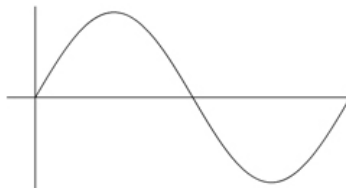
a)



b)

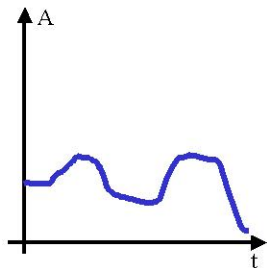


c)

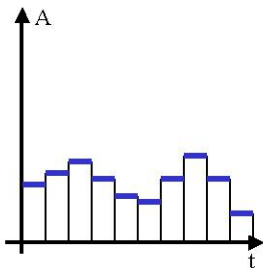


d)

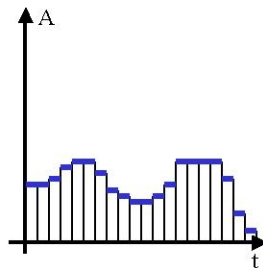
Analógico vs Digital



**Analog signal –
continuously varying**



**Digital signal – large
time divisions**



**Digital signal – small
time divisions**

Funções Analógicas parte 1

Leitura Analógica:

- `analogRead(pino);`
pino = int de 0 a 5, retorna de 0 a 1023.

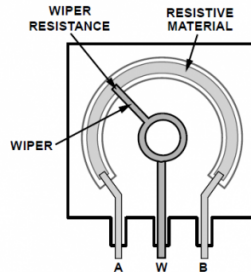
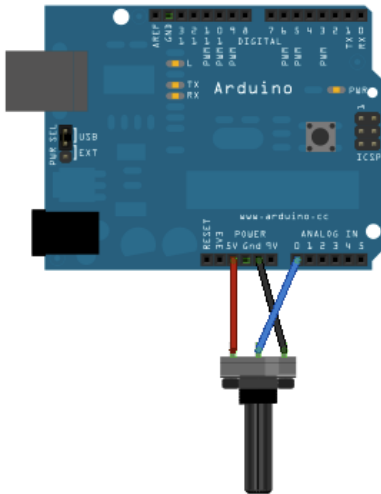
Escrita Analógica:

- Infelizmente o Arduino Uno não conta com essa função. Ao menos não exatamente...

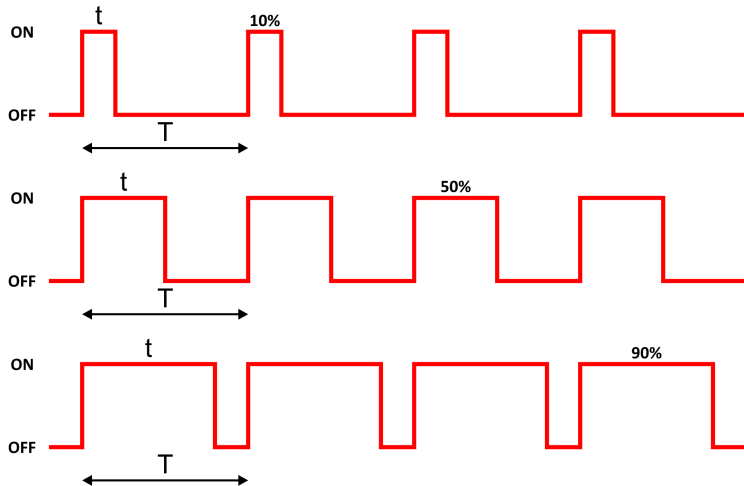
Exercício 02 - Frequência Modificada por Entrada Analógica

```
1 // Frequencia do LED + Porta analogica - Adriano Rodrigues.
2
3 int      led      = 13 ; // Porta do LED.
4 int      periodo  = 0   ; // Valor do period.
5 boolean  estado   = true; // Estado do led.
6
7 void setup()
8 {
9     pinMode(led, OUTPUT); // Configura a porta digital ligada ao LED.
10 }
11
12 void loop()
13 {
14     estado = !estado      ; // Alterna o valor do estado.
15     periodo = analogRead(0) ; // Le o valor do potenciometro.
16     digitalWrite(led,estado);
17     delay(periodo/2)      ; // Periodo determinado pela leitura analogica.
18 }
```

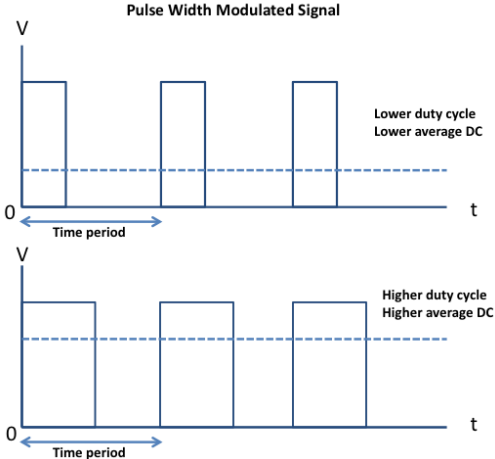
Potenciômetro



PWM - *Pulse Width Modulation* - Modulação por Largura de Pulso



PWM - *Pulse Width Modulation* - Modulação por Largura de Pulso



Leitura Analógica:

- `analogRead(pino);`
pino = int de 0 a 5, retorna de 0 a 1023.

Escrever PWM:

- `analogWrite(pino, valor);`
pino = pinos digitais com '~'. valor = int de 0 a 255.
É necessário definir o pino como saída!

Exercício 03 - Brilho do LED via PWM

Com LED **EXTERNO**!

```
1 // Brilho Variavel do LED via PWM - Adriano Rodrigues.
2 // Porta do LED.
3 int led = 11; // Deve ser uma porta PWM.
4 int buf = 0 ;
5 void setup()
6 {
7     Serial.begin(9600); pinMode(led, OUTPUT);
8 }
9 void loop()
10 {
11     if (Serial.available()>0)
12     {
13         buf = Serial.parseInt(); // Interpreta o dado lido como um int.
14     }
15     analogWrite(led,buf);
16     delay(100);
17 }
```

Tarefa 01 - Brilho Variável do LED via PWM

Com LED **EXTERNO**!

Fazer o brilho do LED aumentar com o tempo.

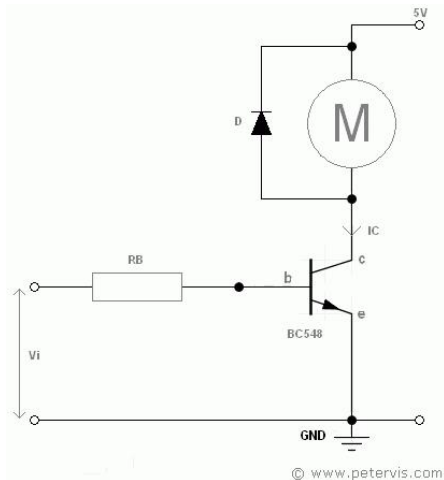
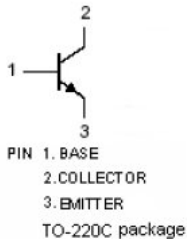
Quando chegar ao brilho máximo, a sequência deve ser recomeçada.

Tarefa 01 - Brilho Variável do LED via PWM

Com LED **EXTERNO**!

```
1 // Brilho Variavel do LED via PWM - Adriano Rodrigues.
2
3 // Porta do LED.
4 int led = 11; // Deve ser uma porta PWM.
5
6 void setup()
7 {
8     // Configuracao da porta digital do LED.
9     pinMode(led, OUTPUT);
10 }
11
12 void loop()
13 {
14     for(int i = 0; i < 255; i++)
15     {
16         analogWrite(led,i);
17         delay(10)      ;
18     }
19 }
```

Acionamento dos Motores



Exercício 04 - Motor DC

```
1 // Velocidade do Motor - Adriano Rodrigues.
2 int comando = 200; // Ciclo de trabalho = 200/255.
3 int port_motor = 3;
4
5 void setup()
6 {
7     pinMode(port_motor,OUTPUT); // Definir como saída.
8 }
9
10 void loop()
11 {
12     analogWrite(port_motor,comando); // Escreve em PWM.
13     delay(100);
14 }
```

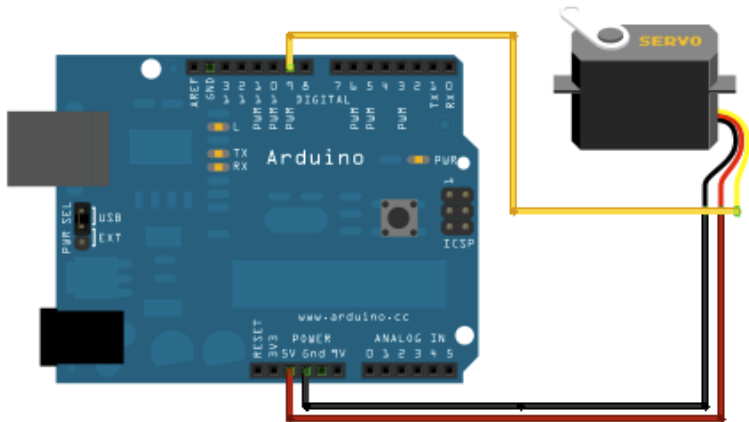
Tarefa 02 - Motor DC com Controle Externo

Controlar a velocidade de um motor manualmente por meio da leitura da tensão de um potenciômetro.

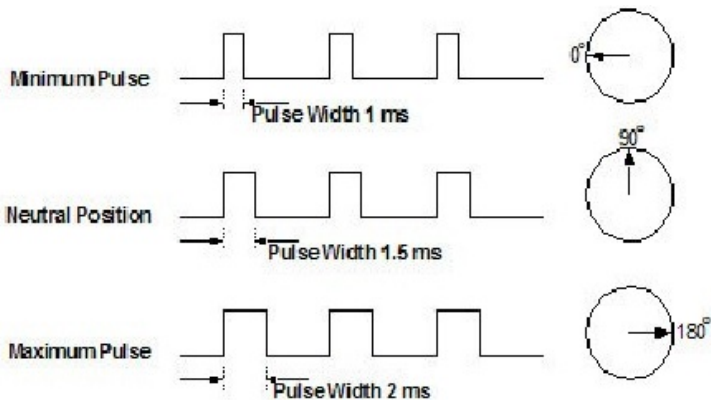
Tarefa 02 - Motor DC com Controle Externo

```
1 // Velocidade do Motor com Potenciometro - Adriano Rodrigues.
2 int regulacao, comando, port_motor = 3;
3
4 void setup()
5 {
6     pinMode(port_motor,OUTPUT); // Definir como saida.
7 }
8
9 void loop()
10 {
11     regulacao = analogRead(A2)      ; // Potenciometro.
12     comando   = regulacao / 4       ; // Mapeia 0-1023 para 0-255.
13     analogWrite(port_motor,comando); // Escreve em PWM.
14     delay(100);
15 }
```

Servomotores



Servomotores



```
#include<biblioteca.h>
```

- `#include<Servo.h>`
- `#include<Wire.h>`
- `#include<Mouse.h>`
- `#include<Keyboard.h>`
- `#include<Stepper.h>`
- `#include<dht.h>`

Exercício 05 - Servo + Biblioteca

```
1 #include <Servo.h> // Adiciona a biblioteca de servos.
2 Servo servo1; // Cria uma variavel servo1 na classe Servo.
3 int pos = 0 ; // Posicao em graus.
4 void setup()
5 {
6     servo1.attach(13); // Anexa o servo a porta 13.
7 }
8 void loop()
9 {
10     for (pos = 0; pos <= 180; pos += 1) // Incrementa o angulo.
11     {
12         servo1.write(pos); // Escreve o angulo desejado no servo.
13         delay(15);
14     }
15     for (pos = 180; pos >= 0; pos -= 1) // Decrementa o angulo.
16     {
17         servo1.write(pos); // Escreve o angulo desejado no servo.
18         delay(15);
19     }
20 }
```

Tarefa 03 - Controlar Ângulo do Servo por Serial

Controlar o ângulo de um motor por meio de comunicação serial.

Tarefa 03 - Controlar Ângulo do Servo por Serial

```
1 // Controlar Angulo do Servo por Serial - Adriano Rodrigues.
2 #include <Servo.h> // Adiciona a biblioteca de servos.
3
4 Servo servo1; // Cria uma variavel servo1 na classe Servo.
5
6 int pos = 0 ; // Posicao em graus.
7
8 void setup()
9 {
10     Serial.begin(9600);
11     servo1.attach(13); // Anexa o servo a porta 13.
12 }
13 void loop()
14 {
15     if (Serial.available() > 0)
16     {
17         pos = Serial.parseInt(); // Interpreta o dado lido como um int.
18     }
19     servo1.write(pos);
20     delay(50);
21 }
```

Até a próxima aula!

Aula 02 - SEEL 2019

Minicurso de Arduino

Comunicação Serial, Portas Analógicas, Bibliotecas e Motores

Adriano Rodrigues

24 de outubro de 2019