

ElectronicRoulette

1.0

Gerado por Doxygen 1.8.18



<b>1 Índice dos Componentes</b>	<b>1</b>
1.1 Lista de Classes	1
<b>2 Índice dos Arquivos</b>	<b>3</b>
2.1 Lista de Arquivos	3
<b>3 Classes</b>	<b>5</b>
3.1 Referência da Estrutura bits_effects_t	5
3.1.1 Descrição detalhada	5
3.2 Referência da Classe ElectronicRoulette	5
3.2.1 Descrição detalhada	6
3.2.2 Construtores e Destrutores	6
3.2.2.1 ElectronicRoulette()	6
3.2.3 Funções membros	6
3.2.3.1 begin()	7
3.2.3.2 printLedsStatus()	7
3.2.3.3 setBuzzerDuration()	7
3.2.3.4 setBuzzerTone()	7
3.2.3.5 setDeceleration()	7
3.2.3.6 setDuration()	8
3.2.3.7 setInitialLedsPins()	8
3.2.3.8 setLedCount()	8
3.2.3.9 setNumbersList()	9
3.2.3.10 setSpeed()	9
3.2.3.11 task()	9
3.2.3.12 test()	9
<b>4 Arquivos</b>	<b>11</b>
4.1 Referência do Arquivo lib/bits_effects/bits_effects.h	11
4.1.1 Descrição detalhada	12
4.1.2 Funções	12
4.1.2.1 bits_effects_all()	12
4.1.2.2 bits_effects_get_bits()	12
4.1.2.3 bits_effects_init()	12
4.1.2.4 bits_effects_reset()	13
4.1.2.5 bits_effects_test()	13
4.2 Referência do Arquivo lib/ElectronicRoulette/ElectronicRoulette.cpp	13
4.2.1 Descrição detalhada	14
4.2.2 Funções	14
4.2.2.1 buttonReadyRoulettePressed()	14
4.2.2.2 buttonStartRoulettePressed()	14
4.2.3 Variáveis	14
4.2.3.1 stateToInterrupt	14

4.3 Referência do Arquivo lib/ElectronicRoulette/ElectronicRoulette.h . . . . .	15
4.3.1 Descrição detalhada . . . . .	16
4.3.2 Enumerações . . . . .	16
4.3.2.1 ElectronicRouletteState . . . . .	16
4.4 Referência do Arquivo src/main.cpp . . . . .	16
4.4.1 Descrição detalhada . . . . .	17
4.4.2 Funções . . . . .	17
4.4.2.1 loop() . . . . .	17
4.4.2.2 setup() . . . . .	17
<b>Índice Remissivo</b>	<b>19</b>

# Capítulo 1

## Índice dos Componentes

### 1.1 Lista de Classes

Aqui estão as classes, estruturas, uniões e interfaces e suas respectivas descrições:

<a href="#">bits_effects_t</a>	Estrutura de dados para controlar os efeitos . . . . .	5
<a href="#">ElectronicRoulette</a>	Classe principal da roleta eletrônica . . . . .	5



## Capítulo 2

# Índice dos Arquivos

### 2.1 Lista de Arquivos

Esta é a lista de todos os arquivos documentados e suas respectivas descrições:

lib/bits_effects/ <a href="#">bits_effects.h</a>	
Biblioteca para controlar efeitos utilizando deslocamento de bits . . . . .	11
lib/ElectronicRoulette/ <a href="#">ElectronicRoulette.cpp</a>	
Classe principal do projeto . . . . .	13
lib/ElectronicRoulette/ <a href="#">ElectronicRoulette.h</a>	
Classe principal do projeto . . . . .	15
src/ <a href="#">main.cpp</a>	
Arquivo principal do código . . . . .	16





## Capítulo 3

# Classes

### 3.1 Referência da Estrutura `bits_effects_t`

Estrutura de dados para controlar os efeitos.

```
#include <bits_effects.h>
```

#### Atributos Públicos

- `uint8_t size`  
*Quantidade de bits até 32, a serem utilizados para os efeitos.*
- `uint8_t speed`  
*Velocidade dos efeitos.*
- `uint32_t bits`  
*Variável para controlar o efeito de 32 bits.*
- `bool effect_done`  
*Efeito concluído.*
- `bool effects_done`  
*Todos os efeitos concluídos.*
- `uint8_t selected_effect`  
*Efeito selecionado.*

#### 3.1.1 Descrição detalhada

Estrutura de dados para controlar os efeitos.

A documentação para essa estrutura foi gerada a partir do seguinte arquivo:

- `lib/bits_effects/bits_effects.h`

### 3.2 Referência da Classe `ElectronicRoulette`

Classe principal da roleta eletrônica.

```
#include <ElectronicRoulette.h>
```

## Membros Públicos

- [ElectronicRoulette](#) ()  
*Constrói um novo objeto Electronic Roulette:: Electronic Roulette.*
- void [begin](#) ()  
*Inicializa a roleta eletrônica.*
- void [task](#) ()  
*Executa as rotinas da roleta eletrônica.*
- void [setInitialLedsPins](#) (uint8\_t initialPin)  
*Define o pino do primeiro led da cadeia de leds da roleta.*
- void [setLedCount](#) (uint8\_t ledCount)  
*Define a quantidade de leds da roleta.*
- void [setSpeed](#) (uint8\_t speed)  
*Define a velocidade da roleta.*
- void [setBuzzerTone](#) (uint16\_t tone)  
*Defini o tom do buzzer.*
- void [setBuzzerDuration](#) (uint8\_t duration)  
*Define a duração do tom do buzzer.*
- void [setDeceleration](#) (uint8\_t deceleration)  
*Define a intensidade de desaceleração da roleta.*
- void [setDuration](#) (uint8\_t duration)  
*Define por quanto tempo a roleta permanece em movimento.*
- void [setNumbersList](#) (uint8\_t numbersList[24])  
*Define a lista de números a serem sorteados.*
- void [test](#) ()  
*Realiza o teste da roleta.*
- void [printLedsStatus](#) ()  
*Imprime o estado dos leds.*

### 3.2.1 Descrição detalhada

Classe principal da roleta eletrônica.

### 3.2.2 Construtores e Destrutores

#### 3.2.2.1 ElectronicRoulette()

```
ElectronicRoulette::ElectronicRoulette ( )
```

Constrói um novo objeto Electronic Roulette:: Electronic Roulette.

Métodos públicos

### 3.2.3 Funções membros

### 3.2.3.1 begin()

```
void ElectronicRoulette::begin ( )
```

Inicializa a roleta eletrônica.

### 3.2.3.2 printLedsStatus()

```
void ElectronicRoulette::printLedsStatus ( )
```

Imprime o estado dos leds.

### 3.2.3.3 setBuzzerDuration()

```
void ElectronicRoulette::setBuzzerDuration (
    uint8_t duration )
```

Define a duração do tom do buzzer.

#### Parâmetros

<i>duration</i>	A duração a ser definida
-----------------	--------------------------

### 3.2.3.4 setBuzzerTone()

```
void ElectronicRoulette::setBuzzerTone (
    uint16_t tone )
```

Defini o tom do buzzer.

#### Parâmetros

<i>tone</i>	O tom a ser definido
-------------	----------------------

### 3.2.3.5 setDeceleration()

```
void ElectronicRoulette::setDeceleration (
    uint8_t deceleration )
```

Define a intensidade de desaceleração da roleta.

**Parâmetros**

<i>deceleration</i>	Intensidade da desaceleração da roleta
---------------------	--

**3.2.3.6 setDuration()**

```
void ElectronicRoulette::setDuration (
    uint8_t duration )
```

Define por quanto tempo a roleta permanece em movimento.

**Parâmetros**

<i>duration</i>	Duração do movimento da roleta
-----------------	--------------------------------

**3.2.3.7 setInitialLedsPins()**

```
void ElectronicRoulette::setInitialLedsPins (
    uint8_t initialPin )
```

Define o pino do primeiro led da cadeia de leds da roleta.

**Parâmetros**

<i>initialPin</i>	Valor do pino
-------------------	---------------

**3.2.3.8 setLedCount()**

```
void ElectronicRoulette::setLedCount (
    uint8_t ledCount )
```

Define a quantidade de leds da roleta.

**Parâmetros**

<i>ledCount</i>	A quantidade de leds
-----------------	----------------------

### 3.2.3.9 setNumbersList()

```
void ElectronicRoulette::setNumbersList (
    uint8_t numbersList[24] )
```

Define a lista de números a serem sorteados.

#### Parâmetros

<i>numbersList</i>	Lista de números de 24 posições
--------------------	---------------------------------

### 3.2.3.10 setSpeed()

```
void ElectronicRoulette::setSpeed (
    uint8_t speed )
```

Define a velocidade da roleta.

#### Parâmetros

<i>speed</i>	Percentual de velocidade de roleta (0 a 100)
--------------	--

### 3.2.3.11 task()

```
void ElectronicRoulette::task ( )
```

Executa as rotinas da roleta eletrônica.

### 3.2.3.12 test()

```
void ElectronicRoulette::test ( )
```

Realiza o teste da roleta.

#### Observação

Necessário remover o método task do loop, para que o teste funcione corretamente

A documentação para essa classe foi gerada a partir dos seguintes arquivos:

- lib/ElectronicRoulette/[ElectronicRoulette.h](#)
- lib/ElectronicRoulette/[ElectronicRoulette.cpp](#)



## Capítulo 4

# Arquivos

### 4.1 Referência do Arquivo lib/bits\_effects/bits\_effects.h

Biblioteca para controlar efeitos utilizando deslocamento de bits.

```
#include <Arduino.h>
```

#### Componentes

- struct [bits\\_effects\\_t](#)  
*Estrutura de dados para controlar os efeitos.*

#### Definições e Macros

- #define **DEFAULT\_MAX\_DELAY** 150
- #define **DEFAULT\_MIN\_DELAY** 30
- #define **EFFECTS\_COUNT** 18

#### Funções

- void [bits\\_effects\\_init](#) ([bits\\_effects\\_t](#) effects\_cfg)  
*Inicializa a biblioteca bits\_effects.*
- bool [bits\\_effects\\_all](#) ()  
*Executa todos os efeitos programados na lista de efeitos.*
- void [bits\\_effects\\_reset](#) ()  
*Reinicia a seleção dos efeitos.*
- uint32\_t [bits\\_effects\\_get\\_bits](#) ()  
*Obtém os valores dos 32 bits processados pela biblioteca.*
- void [bits\\_effects\\_test](#) ()  
*Testa a biblioteca escrevendo a saída do processamento no serial monitor.*

### 4.1.1 Descrição detalhada

Biblioteca para controlar efeitos utilizando deslocamento de bits.

#### Autor

Wesley José Santos (binary-quantum.com)

#### Versão

1.0

#### Data

2020-07-28

#### Copyright

Copyright (c) 2020

### 4.1.2 Funções

#### 4.1.2.1 bits\_effects\_all()

```
bool bits_effects_all ( )
```

Executa todos os efeitos programados na lista de efeitos.

#### Retorna

true Assim que a lista de efeitos é concluída

false Enquanto a lista de efeitos estiver sendo processada

#### 4.1.2.2 bits\_effects\_get\_bits()

```
uint32_t bits_effects_get_bits ( )
```

Obtém os valores dos 32 bits processados pela biblioteca.

#### Retorna

uint32\_t variável contendo os 32 bits processados

#### 4.1.2.3 bits\_effects\_init()

```
void bits_effects_init (
    bits_effects_t effects_cfg )
```

Inicializa a biblioteca bits\_effects.

Funções Públicas



## Parâmetros

<code>effects_cfg</code>	Estrutura de dados com as configurações dos efeitos
--------------------------	---

#### 4.1.2.4 bits\_effects\_reset()

```
void bits_effects_reset ( )
```

Reinicia a seleção dos efeitos.

#### 4.1.2.5 bits\_effects\_test()

```
void bits_effects_test ( )
```

Testa a biblioteca escrevendo a saída do processamento no serial monitor.

## 4.2 Referência do Arquivo lib/ElectronicRoulette/ElectronicRoulette.cpp

Classe principal do projeto.

```
#include "ElectronicRoulette.h"
```

### Funções

- void `buttonReadyRoulettePressed` ()  
*Função a ser chamada pela interrupção quando o botão que prepara a roleta for pressionado.*
- void `buttonStartRoulettePressed` ()  
*Função a ser chamada pela interrupção quando o botão que inicia o sorteio for pressionado.*

### Variáveis

- `ElectronicRouletteState * stateToInterrupt`  
*Estado da roleta eletrônica a ser manipulado pela interrupção.*
- bool `filter` = false  
*Filtro para o botão que aciona os efeitos.*

### 4.2.1 Descrição detalhada

Classe principal do projeto.

#### Autor

Wesley José Santos (binary-quantum.com)

#### Versão

1.0

#### Data

2020-07-27

#### Copyright

Copyright (c) 2020

### 4.2.2 Funções

#### 4.2.2.1 buttonReadyRoulettePressed()

```
void buttonReadyRoulettePressed ( )
```

Função a ser chamada pela interrupção quando o botão que prepara a roleta for pressionado.

Interrupções

#### 4.2.2.2 buttonStartRoulettePressed()

```
void buttonStartRoulettePressed ( )
```

Função a ser chamada pela interrupção quando o botão que inicia o sorteio for pressionado.

### 4.2.3 Variáveis

#### 4.2.3.1 stateToInterrupt

```
ElectronicRouletteState* stateToInterrupt
```

Estado da roleta eletrônica a ser manipulado pela interrupção.

Variáveis globais

## 4.3 Referência do Arquivo lib/ElectronicRoulette/ElectronicRoulette.h

Classe principal do projeto.

```
#include <Arduino.h>
#include "bits_effects.h"
```

### Componentes

- class [ElectronicRoulette](#)  
*Classe principal da roleta eletrônica.*

### Definições e Macros

- #define [DELAY\\_MIN](#) 0  
*Delay máximo para ajuste da velocidade máxima da roleta.*
- #define [DELAY\\_MAX](#) 250  
*Delay mínimo para ajuste da velocidade mínima da roleta.*
- #define [DEFAULT\\_LED\\_COUNT](#) 8  
*Quantidade de leds padrão da roleta.*
- #define [DEFAULT\\_INITIAL\\_SPEED](#) 80  
*Velocidade inicial padrão da roleta.*
- #define [DEFAULT\\_BT\\_RDY\\_PIN](#) 2  
*Pino padrão para o botão que prepara a roleta para o sorteio.*
- #define [DEFAULT\\_BT\\_START\\_PIN](#) 3  
*Pino padrão para o botão que inicia o sorteio da roleta.*
- #define [DEFAULT\\_BUZ\\_PIN](#) 13  
*Pino padrão para conexão do buzzer.*
- #define [DEFAULT\\_INITIAL\\_PIN](#) 4  
*Pino inicial padrão da cadeia de leds da roleta.*
- #define [DEFAULT\\_DECELERATION](#) 3  
*Intensidade inicial padrão da desaceleração da roleta*
- #define [DEFAULT\\_STOP](#) 250  
*Valor da desaceleração necessário para parar a roleta.*
- #define [DEFAULT\\_LIST\\_SIZE](#) 24  
*Valor padrão para o tamanho da lista dos numeros sorteados.*
- #define [DEFAULT\\_BUZZER\\_DURATION](#) 20  
*Valor padrão para a duração do som do buzzer.*
- #define [DEFAULT\\_BUZZER\\_TONE](#) 500  
*Tom padrão do buzzer.*

### Enumerações

- enum [ElectronicRouletteState](#) { [ST\\_IDLE](#), [ST\\_READY](#), [ST\\_DRAWING](#), [ST\\_DRAWN](#) }  
*Estados da roleta eletrônica.*

### 4.3.1 Descrição detalhada

Classe principal do projeto.

**Autor**

Wesley José Santos (binary-quantum.com)

**Versão**

1.0

**Data**

2020-07-27

**Copyright**

Copyright (c) 2020

### 4.3.2 Enumerações

#### 4.3.2.1 ElectronicRouletteState

enum `ElectronicRouletteState`

Estados da roleta eletrônica.

**Enumeradores**

ST_IDLE	Aguardando comando, reproduzindo efeitos nos leds.
ST_READY	Aguardando comando, leds apagados.
ST_DRAWING	Realizando sorteio.
ST_DRAWN	Sorteio realizado. Aguardando comando.

## 4.4 Referência do Arquivo src/main.cpp

Arquivo principal do código.

```
#include <Arduino.h>
#include "ElectronicRoulette.h"
```

## Funções

- void `setup` ()  
*Setup principal do Arduino.*
- void `loop` ()  
*Loop principal do Arduino.*

## Variáveis

- `ElectronicRoulette` *roleta*  
*Instância global da roleta.*
- `uint8_t numerosDaSorte` [24] = {3, 2, 4, 2, 3, 4, 1, 5, 6, 3, 7, 2, 1, 5, 2, 5, 3, 6, 5, 4, 1, 8, 3, 1}  
*Lista de números a serem sorteados.*

### 4.4.1 Descrição detalhada

Arquivo principal do código.

#### Autor

Wesley José Santos (binary-quantum.com)

#### Versão

1.0

#### Data

2020-07-28

#### Copyright

Copyright (c) 2020

### 4.4.2 Funções

#### 4.4.2.1 `loop()`

```
void loop ( )
```

Loop principal do Arduino.

#### 4.4.2.2 `setup()`

```
void setup ( )
```

Setup principal do Arduino.



# Índice Remissivo

- begin
  - ElectronicRoulette, 6
- bits\_effects.h
  - bits\_effects\_all, 12
  - bits\_effects\_get\_bits, 12
  - bits\_effects\_init, 12
  - bits\_effects\_reset, 13
  - bits\_effects\_test, 13
- bits\_effects\_all
  - bits\_effects.h, 12
- bits\_effects\_get\_bits
  - bits\_effects.h, 12
- bits\_effects\_init
  - bits\_effects.h, 12
- bits\_effects\_reset
  - bits\_effects.h, 13
- bits\_effects\_t, 5
- bits\_effects\_test
  - bits\_effects.h, 13
- buttonReadyRoulettePressed
  - ElectronicRoulette.cpp, 14
- buttonStartRoulettePressed
  - ElectronicRoulette.cpp, 14
- ElectronicRoulette, 5
  - begin, 6
  - ElectronicRoulette, 6
  - printLedsStatus, 7
  - setBuzzerDuration, 7
  - setBuzzerTone, 7
  - setDeceleration, 7
  - setDuration, 8
  - setInitialLedsPins, 8
  - setLedCount, 8
  - setNumbersList, 8
  - setSpeed, 9
  - task, 9
  - test, 9
- ElectronicRoulette.cpp
  - buttonReadyRoulettePressed, 14
  - buttonStartRoulettePressed, 14
  - stateToInterrupt, 14
- ElectronicRoulette.h
  - ElectronicRouletteState, 16
  - ST\_DRAWING, 16
  - ST\_DRAWN, 16
  - ST\_IDLE, 16
  - ST\_READY, 16
- ElectronicRouletteState
  - ElectronicRoulette.h, 16
- lib/bits\_effects/bits\_effects.h, 11
- lib/ElectronicRoulette/ElectronicRoulette.cpp, 13
- lib/ElectronicRoulette/ElectronicRoulette.h, 15
- loop
  - main.cpp, 17
- main.cpp
  - loop, 17
  - setup, 17
- printLedsStatus
  - ElectronicRoulette, 7
- setBuzzerDuration
  - ElectronicRoulette, 7
- setBuzzerTone
  - ElectronicRoulette, 7
- setDeceleration
  - ElectronicRoulette, 7
- setDuration
  - ElectronicRoulette, 8
- setInitialLedsPins
  - ElectronicRoulette, 8
- setLedCount
  - ElectronicRoulette, 8
- setNumbersList
  - ElectronicRoulette, 8
- setSpeed
  - ElectronicRoulette, 9
- setup
  - main.cpp, 17
- src/main.cpp, 16
- ST\_DRAWING
  - ElectronicRoulette.h, 16
- ST\_DRAWN
  - ElectronicRoulette.h, 16
- ST\_IDLE
  - ElectronicRoulette.h, 16
- ST\_READY
  - ElectronicRoulette.h, 16
- stateToInterrupt
  - ElectronicRoulette.cpp, 14
- task
  - ElectronicRoulette, 9
- test
  - ElectronicRoulette, 9