

StepMotor library for Arduino & ARM Cortex-M3 (STM32)

Generated by Doxygen 1.8.17



<b>1 Class Index</b>	<b>1</b>
1.1 Class List	1
<b>2 File Index</b>	<b>3</b>
2.1 File List	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 StepMotor Class Reference	5
3.1.1 Constructor & Destructor Documentation	6
3.1.1.1 StepMotor()	6
3.1.2 Member Function Documentation	6
3.1.2.1 _BI_CLKWISE_ROT_FULL_TORQUE()	6
3.1.2.2 _BI_CTR_CLKWISE_ROT_FULL_TORQUE()	6
3.1.2.3 _RELEASE_PINS()	7
3.1.2.4 _UNI_CLKWISE_ROT_FULL_TORQUE()	7
3.1.2.5 _UNI_CTR_CLKWISE_ROT_FULL_TORQUE()	7
3.1.2.6 begin()	7
3.1.2.7 end()	7
3.1.2.8 setMotorType()	7
3.1.2.9 setMov()	8
3.1.2.10 setTorqueForce()	8
3.1.3 Member Data Documentation	8
3.1.3.1 _in1Pin	8
3.1.3.2 _in2Pin	8
3.1.3.3 _in3Pin	8
3.1.3.4 _in4Pin	9
3.1.3.5 _motorT	9
3.1.3.6 _torqueF	9
<b>4 File Documentation</b>	<b>11</b>
4.1 StepMotor.cpp File Reference	11
4.2 StepMotor.h File Reference	11
4.2.1 Macro Definition Documentation	13
4.2.1.1 IN1_HIGH	13
4.2.1.2 IN1_INPUT	13
4.2.1.3 IN1_LOW	13
4.2.1.4 IN1_OUTPUT	13
4.2.1.5 IN2_HIGH	13
4.2.1.6 IN2_INPUT	13
4.2.1.7 IN2_LOW	14
4.2.1.8 IN2_OUTPUT	14
4.2.1.9 IN3_HIGH	14
4.2.1.10 IN3_INPUT	14

4.2.1.11 IN3_LOW . . . . .	14
4.2.1.12 IN3_OUTPUT . . . . .	14
4.2.1.13 IN4_HIGH . . . . .	14
4.2.1.14 IN4_INPUT . . . . .	14
4.2.1.15 IN4_LOW . . . . .	15
4.2.1.16 IN4_OUTPUT . . . . .	15
4.2.2 Enumeration Type Documentation . . . . .	15
4.2.2.1 directRot . . . . .	15
4.2.2.2 motorType . . . . .	15
4.2.2.3 torqueForce . . . . .	15
<b>Index</b>	<b>17</b>

# Chapter 1

## Class Index

### 1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">StepMotor</a> . . . . .	5
-------------------------------------	---



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">StepMotor.cpp</a>	.....	11
<a href="#">StepMotor.h</a>	.....	11





## Chapter 3

# Class Documentation

### 3.1 StepMotor Class Reference

```
#include <StepMotor.h>
```

#### Public Member Functions

- [StepMotor](#) (uint8\_t in1Pin, uint8\_t in2Pin, uint8\_t in3Pin, uint8\_t in4Pin, [motorType](#) motorT, [torqueForce](#) torqueF=[Full](#))
- virtual void [begin](#) ()  
*Initialize [StepMotor](#) controller.*
- virtual void [end](#) ()  
*Release [StepMotor](#) controller pins.*
- virtual void [setMov](#) (uint16\_t nSteps, uint8\_t speed=1, [directRot](#) directR=[Clk](#))  
*Rotate [StepMotor](#) nSteps at the established speed and direction.*
- virtual void [setMotorType](#) ([motorType](#) motorT)  
*Set which motor type will be used (Bipolar or Unipolar)*
- virtual void [setTorqueForce](#) ([torqueForce](#) torqueF)  
*Set which torque force will be used (Half or Full)*

#### Protected Member Functions

- void [\\_RELEASE\\_PINS](#) ()
- void [\\_BI\\_CLKWISE\\_ROT\\_FULL\\_TORQUE](#) (uint16\_t nSteps, uint8\_t speed)  
*Clockwise Rotation of Bipolar Motor at Full Torque.*
- void [\\_BI\\_CTR\\_CLKWISE\\_ROT\\_FULL\\_TORQUE](#) (uint16\_t nSteps, uint8\_t speed)  
*Counter Clockwise Rotation of Bipolar Motor at Full Torque.*
- void [\\_UNI\\_CLKWISE\\_ROT\\_FULL\\_TORQUE](#) (uint16\_t nSteps, uint8\_t speed)  
*Clockwise Rotation of Unipolar Motor at Full Torque.*
- void [\\_UNI\\_CTR\\_CLKWISE\\_ROT\\_FULL\\_TORQUE](#) (uint16\_t nSteps, uint8\_t speed)  
*Counter Clockwise Rotation of Unipolar Motor at Full Torque.*

## Protected Attributes

- [uint8\\_t \\_in1Pin](#)  
*Pole 1 pin.*
- [uint8\\_t \\_in2Pin](#)  
*Pole 2 pin.*
- [uint8\\_t \\_in3Pin](#)  
*Pole 3 pin.*
- [uint8\\_t \\_in4Pin](#)  
*Pole 4 pin.*
- [motorType \\_motorT](#)  
*Motor type: Bipolar or Unipolar.*
- [torqueForce \\_torqueF](#)  
*Torque force: Full or Half.*

## 3.1.1 Constructor & Destructor Documentation

### 3.1.1.1 StepMotor()

```
StepMotor::StepMotor (
    uint8_t in1Pin,
    uint8_t in2Pin,
    uint8_t in3Pin,
    uint8_t in4Pin,
    motorType motorT,
    torqueForce torqueF = Full )
```

## 3.1.2 Member Function Documentation

### 3.1.2.1 \_BI\_CLKWISE\_ROT\_FULL\_TORQUE()

```
void StepMotor::_BI_CLKWISE_ROT_FULL_TORQUE (
    uint16_t nSteps,
    uint8_t speed ) [protected]
```

Clockwise Rotation of Bipolar Motor at Full Torque.

### 3.1.2.2 \_BI\_CTR\_CLKWISE\_ROT\_FULL\_TORQUE()

```
void StepMotor::_BI_CTR_CLKWISE_ROT_FULL_TORQUE (
    uint16_t nSteps,
    uint8_t speed ) [protected]
```

Counter Clockwise Rotation of Bipolar Motor at Full Torque.

### 3.1.2.3 \_RELEASE\_PINS()

```
void StepMotor::_RELEASE_PINS ( ) [protected]
```

### 3.1.2.4 \_UNI\_CLKWISE\_ROT\_FULL\_TORQUE()

```
void StepMotor::_UNI_CLKWISE_ROT_FULL_TORQUE (
    uint16_t nSteps,
    uint8_t speed ) [protected]
```

Clockwise Rotation of Unipolar Motor at Full Torque.

### 3.1.2.5 \_UNI\_CTR\_CLKWISE\_ROT\_FULL\_TORQUE()

```
void StepMotor::_UNI_CTR_CLKWISE_ROT_FULL_TORQUE (
    uint16_t nSteps,
    uint8_t speed ) [protected]
```

Counter Clockwise Rotation of Unipolar Motor at Full Torque.

### 3.1.2.6 begin()

```
void StepMotor::begin ( ) [virtual]
```

Initialize [StepMotor](#) controller.

### 3.1.2.7 end()

```
void StepMotor::end ( ) [virtual]
```

Release [StepMotor](#) controller pins.

### 3.1.2.8 setMotorType()

```
void StepMotor::setMotorType (
    motorType motorT ) [virtual]
```

Set which motor type will be used (Bipolar or Unipolar)

### 3.1.2.9 setMov()

```
void StepMotor::setMov (
    uint16_t nSteps,
    uint8_t speed = 1,
    directRot directR = Clk ) [virtual]
```

Rotate [StepMotor](#) nSteps at the established speed and direction.

### 3.1.2.10 setTorqueForce()

```
void StepMotor::setTorqueForce (
    torqueForce torqueF ) [virtual]
```

Set which torque force will be used (Half or Full)

## 3.1.3 Member Data Documentation

### 3.1.3.1 \_in1Pin

```
uint8_t StepMotor::_in1Pin [protected]
```

Pole 1 pin.

### 3.1.3.2 \_in2Pin

```
uint8_t StepMotor::_in2Pin [protected]
```

Pole 2 pin.

### 3.1.3.3 \_in3Pin

```
uint8_t StepMotor::_in3Pin [protected]
```

Pole 3 pin.

#### 3.1.3.4 \_in4Pin

```
uint8_t StepMotor::_in4Pin [protected]
```

Pole 4 pin.

#### 3.1.3.5 \_motorT

```
motorType StepMotor::_motorT [protected]
```

Motor type: Bipolar or Unipolar.

#### 3.1.3.6 \_torqueF

```
torqueForce StepMotor::_torqueF [protected]
```

Torque force: Full or Half.

The documentation for this class was generated from the following files:

- [StepMotor.h](#)
- [StepMotor.cpp](#)



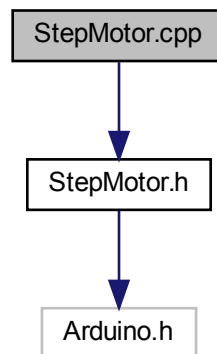
## Chapter 4

# File Documentation

### 4.1 StepMotor.cpp File Reference

```
#include "StepMotor.h"
```

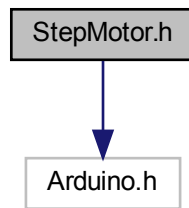
Include dependency graph for StepMotor.cpp:



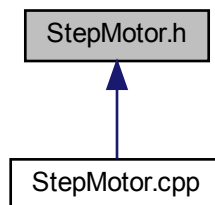
### 4.2 StepMotor.h File Reference

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

Include dependency graph for StepMotor.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [StepMotor](#)

## Macros

- `#define IN1_LOW() { digitalWrite(_in1Pin, LOW); }`
- `#define IN1_HIGH() { digitalWrite(_in1Pin, HIGH); }`
- `#define IN1_INPUT() { pinMode(_in1Pin, INPUT); }`
- `#define IN1_OUTPUT() { pinMode(_in1Pin, OUTPUT); }`
- `#define IN2_LOW() { digitalWrite(_in2Pin, LOW); }`
- `#define IN2_HIGH() { digitalWrite(_in2Pin, HIGH); }`
- `#define IN2_INPUT() { pinMode(_in2Pin, INPUT); }`
- `#define IN2_OUTPUT() { pinMode(_in2Pin, OUTPUT); }`
- `#define IN3_LOW() { digitalWrite(_in3Pin, LOW); }`
- `#define IN3_HIGH() { digitalWrite(_in3Pin, HIGH); }`
- `#define IN3_INPUT() { pinMode(_in3Pin, INPUT); }`
- `#define IN3_OUTPUT() { pinMode(_in3Pin, OUTPUT); }`
- `#define IN4_LOW() { digitalWrite(_in4Pin, LOW); }`
- `#define IN4_HIGH() { digitalWrite(_in4Pin, HIGH); }`
- `#define IN4_INPUT() { pinMode(_in4Pin, INPUT); }`
- `#define IN4_OUTPUT() { pinMode(_in4Pin, OUTPUT); }`



## Enumerations

- enum `motorType` { `Unipolar`, `Bipolar` }
- enum `torqueForce` { `Half`, `Full` }
- enum `directRot` { `Clk`, `CtrClk` }

## 4.2.1 Macro Definition Documentation

### 4.2.1.1 IN1\_HIGH

```
#define IN1_HIGH( ) { digitalWrite(_in1Pin, HIGH);}
```

### 4.2.1.2 IN1\_INPUT

```
#define IN1_INPUT( ) { pinMode( _in1Pin, INPUT); }
```

### 4.2.1.3 IN1\_LOW

```
#define IN1_LOW( ) { digitalWrite(_in1Pin, LOW); }
```

### 4.2.1.4 IN1\_OUTPUT

```
#define IN1_OUTPUT( ) { pinMode( _in1Pin, OUTPUT);}
```

### 4.2.1.5 IN2\_HIGH

```
#define IN2_HIGH( ) { digitalWrite(_in2Pin, HIGH);}
```

### 4.2.1.6 IN2\_INPUT

```
#define IN2_INPUT( ) { pinMode( _in2Pin, INPUT); }
```

#### 4.2.1.7 IN2\_LOW

```
#define IN2_LOW( ) { digitalWrite(_in2Pin, LOW); }
```

#### 4.2.1.8 IN2\_OUTPUT

```
#define IN2_OUTPUT( ) { pinMode( _in2Pin, OUTPUT);}
```

#### 4.2.1.9 IN3\_HIGH

```
#define IN3_HIGH( ) { digitalWrite(_in3Pin, HIGH);}
```

#### 4.2.1.10 IN3\_INPUT

```
#define IN3_INPUT( ) { pinMode( _in3Pin, INPUT); }
```

#### 4.2.1.11 IN3\_LOW

```
#define IN3_LOW( ) { digitalWrite(_in3Pin, LOW); }
```

#### 4.2.1.12 IN3\_OUTPUT

```
#define IN3_OUTPUT( ) { pinMode( _in3Pin, OUTPUT);}
```

#### 4.2.1.13 IN4\_HIGH

```
#define IN4_HIGH( ) { digitalWrite(_in4Pin, HIGH);}
```

#### 4.2.1.14 IN4\_INPUT

```
#define IN4_INPUT( ) { pinMode( _in4Pin, INPUT); }
```

#### 4.2.1.15 IN4\_LOW

```
#define IN4_LOW( ) { digitalWrite(_in4Pin, LOW); }
```

#### 4.2.1.16 IN4\_OUTPUT

```
#define IN4_OUTPUT( ) { pinMode( _in4Pin, OUTPUT); }
```

### 4.2.2 Enumeration Type Documentation

#### 4.2.2.1 directRot

```
enum directRot
```

Enumerator

Clk	
CtrClk	

#### 4.2.2.2 motorType

```
enum motorType
```

Enumerator

Unipolar	
Bipolar	

#### 4.2.2.3 torqueForce

```
enum torqueForce
```

Enumerator

Half	
Full	



# Index

`_BI_CLKWISE_ROT_FULL_TORQUE`  
    StepMotor, [6](#)

`_BI_CTR_CLKWISE_ROT_FULL_TORQUE`  
    StepMotor, [6](#)

`_RELEASE_PINS`  
    StepMotor, [6](#)

`_UNI_CLKWISE_ROT_FULL_TORQUE`  
    StepMotor, [7](#)

`_UNI_CTR_CLKWISE_ROT_FULL_TORQUE`  
    StepMotor, [7](#)

`_in1Pin`  
    StepMotor, [8](#)

`_in2Pin`  
    StepMotor, [8](#)

`_in3Pin`  
    StepMotor, [8](#)

`_in4Pin`  
    StepMotor, [8](#)

`_motorT`  
    StepMotor, [9](#)

`_torqueF`  
    StepMotor, [9](#)

`begin`  
    StepMotor, [7](#)

`Bipolar`  
    StepMotor.h, [15](#)

`Clk`  
    StepMotor.h, [15](#)

`CtrClk`  
    StepMotor.h, [15](#)

`directRot`  
    StepMotor.h, [15](#)

`end`  
    StepMotor, [7](#)

`Full`  
    StepMotor.h, [15](#)

`Half`  
    StepMotor.h, [15](#)

`IN1_HIGH`  
    StepMotor.h, [13](#)

`IN1_INPUT`  
    StepMotor.h, [13](#)

`IN1_LOW`  
    StepMotor.h, [13](#)

`IN1_OUTPUT`  
    StepMotor.h, [13](#)

`IN2_HIGH`  
    StepMotor.h, [13](#)

`IN2_INPUT`  
    StepMotor.h, [13](#)

`IN2_LOW`  
    StepMotor.h, [13](#)

`IN2_OUTPUT`  
    StepMotor.h, [14](#)

`IN3_HIGH`  
    StepMotor.h, [14](#)

`IN3_INPUT`  
    StepMotor.h, [14](#)

`IN3_LOW`  
    StepMotor.h, [14](#)

`IN3_OUTPUT`  
    StepMotor.h, [14](#)

`IN4_HIGH`  
    StepMotor.h, [14](#)

`IN4_INPUT`  
    StepMotor.h, [14](#)

`IN4_LOW`  
    StepMotor.h, [14](#)

`IN4_OUTPUT`  
    StepMotor.h, [15](#)

`motorType`  
    StepMotor.h, [15](#)

`setMotorType`  
    StepMotor, [7](#)

`setMov`  
    StepMotor, [7](#)

`setTorqueForce`  
    StepMotor, [8](#)

`StepMotor`, [5](#)  
    [\\_BI\\_CLKWISE\\_ROT\\_FULL\\_TORQUE](#), [6](#)  
    [\\_BI\\_CTR\\_CLKWISE\\_ROT\\_FULL\\_TORQUE](#), [6](#)  
    [\\_RELEASE\\_PINS](#), [6](#)  
    [\\_UNI\\_CLKWISE\\_ROT\\_FULL\\_TORQUE](#), [7](#)  
    [\\_UNI\\_CTR\\_CLKWISE\\_ROT\\_FULL\\_TORQUE](#), [7](#)  
    [\\_in1Pin](#), [8](#)  
    [\\_in2Pin](#), [8](#)  
    [\\_in3Pin](#), [8](#)  
    [\\_in4Pin](#), [8](#)  
    [\\_motorT](#), [9](#)  
    [\\_torqueF](#), [9](#)  
    [begin](#), [7](#)  
    [end](#), [7](#)

- setMotorType, [7](#)
- setMov, [7](#)
- setTorqueForce, [8](#)
- StepMotor, [6](#)
- StepMotor.cpp, [11](#)
- StepMotor.h, [11](#)
  - Bipolar, [15](#)
  - Clk, [15](#)
  - CtrClk, [15](#)
  - directRot, [15](#)
  - Full, [15](#)
  - Half, [15](#)
  - IN1\_HIGH, [13](#)
  - IN1\_INPUT, [13](#)
  - IN1\_LOW, [13](#)
  - IN1\_OUTPUT, [13](#)
  - IN2\_HIGH, [13](#)
  - IN2\_INPUT, [13](#)
  - IN2\_LOW, [13](#)
  - IN2\_OUTPUT, [14](#)
  - IN3\_HIGH, [14](#)
  - IN3\_INPUT, [14](#)
  - IN3\_LOW, [14](#)
  - IN3\_OUTPUT, [14](#)
  - IN4\_HIGH, [14](#)
  - IN4\_INPUT, [14](#)
  - IN4\_LOW, [14](#)
  - IN4\_OUTPUT, [15](#)
  - motorType, [15](#)
  - torqueForce, [15](#)
  - Unipolar, [15](#)
- torqueForce
  - StepMotor.h, [15](#)
- Unipolar
  - StepMotor.h, [15](#)