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

Generated by Doxygen 1.8.17

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

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

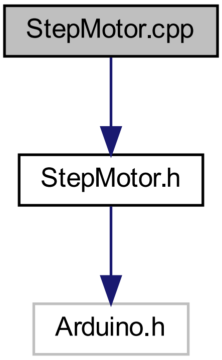
Chapter 4

File Documentation

4.1 StepMotor.cpp File Reference

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

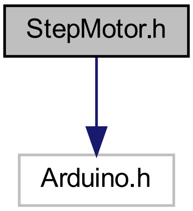
Include dependency graph for StepMotor.cpp:



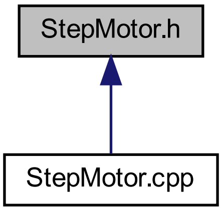
4.2 StepMotor.h File Reference

```
#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