Sabertooth Simplified Serial Library for Arduino

Control your Sabertooth with Simplified Serial.

Class List

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

| SabertoothSimplified | Controls a Sabertooth motor driver running in Simplified Serial mode |
Sabertooth Simplified Serial Library for Arduino

Control your Sabertooth with Simplified Serial.

SabertoothSimplified
Class Reference

Controls a Sabertooth motor driver running in Simplified Serial mode. More...
### Public Member Functions

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SabertoothSimplified</strong>()</td>
<td></td>
</tr>
<tr>
<td><strong>SabertoothSimplified</strong>(Print &amp;port)</td>
<td></td>
</tr>
<tr>
<td>void <strong>motor</strong>(int power)</td>
<td></td>
</tr>
<tr>
<td>void <strong>motor</strong>(byte motor, int power)</td>
<td></td>
</tr>
<tr>
<td>void <strong>drive</strong>(int power)</td>
<td></td>
</tr>
<tr>
<td>void <strong>turn</strong>(int power)</td>
<td></td>
</tr>
<tr>
<td>void <strong>stop</strong>()</td>
<td></td>
</tr>
</tbody>
</table>
Detailed Description

Controls a Sabertooth motor driver running in Simplified Serial mode.

Examples:
- SimpleExample/SimpleExample.ino,
- SoftwareSerial/SoftwareSerial.ino, Sweep/Sweep.ino,
- and TankStyleSweep.ino.
Constructor & Destructor Documentation

**SabertoothSimplified::SabertoothSimplified ( )**

Initializes a new instance of the `SabertoothSimplified` class. The Arduino TX serial port is used.

**SabertoothSimplified::SabertoothSimplified ( Print & port**

Initializes a new instance of the `SabertoothSimplified` class. The specified serial port is used.

**Parameters**

- **port** The port to use.
Member Function Documentation

```cpp
void SabertoothSimplified::drive (int power)
```

Sets the driving power.

**Parameters**

*power* The power, between -127 and 127.

**Examples:**

*TankStyleSweep.ino.*

```cpp
void SabertoothSimplified::motor (int power)
```

Sets the power of motor 1.

**Parameters**

*power* The power, between -127 and 127.

**Examples:**

*SimpleExample/SimpleExample.ino*, and *Sweep/Sweep.ino.*

```cpp
void SabertoothSimplified::motor (byte motor, int power)
```

Sets the power of the specified motor.

**Parameters**
- `motor` The motor number, 1 or 2.
- `power` The power, between -127 and 127.

```
void SabertoothSimplified::stop()
```

Stops.

```
void SabertoothSimplified::turn(int power)
```

Sets the turning power.

**Parameters**
- `power` The power, between -127 and 127.

**Examples:**
- `TankStyleSweep.ino`. 
Sabertooth Simplified Serial Library for Arduino

Control your Sabertooth with Simplified Serial.

<table>
<thead>
<tr>
<th>Main Page</th>
<th>Classes</th>
<th>Files</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class List</td>
<td>Class Index</td>
<td>Class Members</td>
<td></td>
</tr>
</tbody>
</table>

### Class Index

<table>
<thead>
<tr>
<th>S</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>S</th>
</tr>
</thead>
</table>

SabertoothSimplified
Sabertooth Simplified Serial Library for Arduino

Control your Sabertooth with Simplified Serial.

Here is a list of all documented class members with links to the class documentation for each member:

- `drive()` : `SabertoothSimplified`
- `motor()` : `SabertoothSimplified`
- `SabertoothSimplified()` : `SabertoothSimplified`
- `stop()` : `SabertoothSimplified`
- `turn()` : `SabertoothSimplified`
Sabertooth Simplified Serial Library for Arduino

Control your Sabertooth with Simplified Serial.

- `drive()` : `SabertoothSimplified`
- `motor()` : `SabertoothSimplified`
- `SabertoothSimplified()` : `SabertoothSimplified`
- `stop()` : `SabertoothSimplified`
- `turn()` : `SabertoothSimplified`
# Sabertooth Simplified Serial Library for Arduino

Control your Sabertooth with Simplified Serial.

<table>
<thead>
<tr>
<th>Main Page</th>
<th>Classes</th>
<th>Files</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>File List</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## File List

Here is a list of all documented files with brief descriptions:

- **SabertoothSimplified**
- **SabertoothSimplified.h**
# Sabertooth Simplified Serial Library for Arduino

Control your Sabertooth with Simplified Serial.

<table>
<thead>
<tr>
<th>Main Page</th>
<th>Classes</th>
<th>Files</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>SabertoothSimplified</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SabertoothSimplified Directory Reference**
### Files

<table>
<thead>
<tr>
<th>file</th>
<th>SabertoothSimplified.cpp</th>
</tr>
</thead>
<tbody>
<tr>
<td>file</td>
<td>SabertoothSimplified.h</td>
</tr>
</tbody>
</table>
Sabertooth Simplified Serial Library for Arduino

Control your Sabertooth with Simplified Serial.

### SabertoothSimplified.h

```cpp
/*
 * Arduino Library for Sabertooth Simplified Serial
 * Copyright (c) 2012-2013 Dimension Engineering LLC
 * http://www.dimensionengineering.com/arduino
 * Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.
 * THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
```
#ifndef SabertoothSimplified_h
#define SabertoothSimplified_h

#if defined(ARDUINO) && ARDUINO >= 100
#include <Arduino.h>
#else
#include <WProgram.h>
#endif

#if defined(USBCON)
#define SabertoothTXPinSerial Serial1
  // Arduino Leonardo has TX->1 on Serial1, not Serial.
#else
#define SabertoothTXPinSerial Serial
#endif
#define SyRenTXPinSerial SabertoothTXPinSerial

class SabertoothSimplified
{
public:
  SabertoothSimplified();
  SabertoothSimplified(Print& port);

public:
  void motor(int power);
  void motor(byte motor, int power);
  void drive(int power);
void turn(int power);

void stop();

private:
void mixedMode(boolean enable);
void mixedUpdate();
void raw(byte motor, int power);

private:
boolean _mixed;
int _mixedDrive, _mixedTurn;
boolean _mixedDriveSet, _mixedTurnSet;
Print& _port;

};

#endif
Sabertooth Simplified Serial Library for Arduino
Control your Sabertooth with Simplified Serial.

Examples

Here is a list of all examples:

- SimpleExample/SimpleExample.ino
- SoftwareSerial/SoftwareSerial.ino
- Sweep/Sweep.ino
- TankStyleSweep.ino
Sabertooth Simplified Serial Library for Arduino

Control your Sabertooth with Simplified Serial.

SimpleExample/SimpleExample.ino

Goes in one direction, stops, and then goes in the other direction.

```
// Simple Example Sample
// Copyright (c) 2012 Dimension Engineering LLC
// See license.txt for license details.

#include <SabertoothSimplified.h>

SabertoothSimplified ST; // We'll name the Sabertooth object ST.
// For how to configure the Sabertooth, see the DIP Switch Wizard for
// http://www.dimensionengineering.com/datasheets/SabertoothDIPWizard/start.htm
// Be sure to select Simplified Serial Mode for use with this library.
// This sample uses a baud rate of 9600.
//
// Connections to make:
//  Arduino TX->1   ->  Sabertooth S1
//  Arduino GND    ->  Sabertooth 0V
//  Arduino VIN    ->  Sabertooth 5V (OPTIONAL, if you want the Sabertooth to power the Arduino)
```
// If you want to use a pin other than TX->1, see the SoftwareSerial example.

void setup()
{
    SabertoothTXPinSerial.begin(9600); // This is the baud rate you chose with the DIP switches.
}

void loop()
{
    ST.motor(1, 127); // Go forward at full power.
    delay(2000); // Wait 2 seconds.
    ST.motor(1, 0); // Stop.
    delay(2000); // Wait 2 seconds.
    ST.motor(1, -127); // Reverse at full power.
    delay(2000); // Wait 2 seconds.
    ST.motor(1, 0); // Stop.
    delay(2000);
}
Sabertooth Simplified Serial Library for Arduino
Control your Sabertooth with Simplified Serial.

SoftwareSerial/SoftwareSerial.ino

Uses a pin other than TX to connect to S1.

```cpp
// Software Serial Sample
// Copyright (c) 2012 Dimension Engineering LLC
// See license.txt for license details.

#include <SoftwareSerial.h>
#include <SabertoothSimplified.h>

SoftwareSerial SWSerial(NOT_A_PIN, 11); // RX on no pin (unused), TX on pin 11 (to S1).
SabertoothSimplified ST(SWSerial); // Use SWSerial as the serial port.

void setup()
{
  SWSerial.begin(9600);
}

void loop()
{
  int power;

  // Ramp from -127 to 127 (full reverse to full forward), waiting 20 ms (1/50th of a second) per value.
  for (power = -127; power <= 127; power ++)
  {
  
```
ST.motor(1, power);
delay(20);
}

// Now go back the way we came.
for (power = 127; power >= -127; power --)
{
    ST.motor(1, power);
delay(20);
}
Sabertooth Simplified Serial Library for Arduino

Control your Sabertooth with Simplified Serial.

Sweep/Sweep.ino

Sweeps from full reverse to full forward and then from full forward to full reverse.

```cpp
// Sweep Sample
// Copyright (c) 2012 Dimension Engineering LLC
// See license.txt for license details.

#include <SabertoothSimplified.h>

SabertoothSimplified ST; // We'll name the Sabertooth object ST.

// For how to configure the Sabertooth, see the DIP Switch Wizard for

// http://www.dimensionengineering.com/datasheets/SabertoothDIPWizard/start.htm

// Be sure to select Simplified Serial Mode for use with this library.

// This sample uses a baud rate of 9600.

// Connections to make:

// Arduino TX->1 -> Sabertooth S1
// Arduino GND      -> Sabertooth 0V
// Arduino VIN     -> Sabertooth 5V (OPTIONAL, if you want the Sabertooth to power the Arduino)
```
// If you want to use a pin other than TX->1, see the SoftwareSerial example.

void setup()
{
    SabertoothTXPinSerial.begin(9600); // This is the baud rate you chose with the DIP switches.
}

void loop()
{
    int power;

    // Ramp motor 1 and motor 2 from -127 to 127 (full reverse to full forward),
    // waiting 20 ms (1/50th of a second) per value.
    for (power = -127; power <= 127; power ++)
    {
        ST.motor(1, power);
        ST.motor(2, power);
        delay(20);
    }

    // Now go back the way we came.
    for (power = 127; power >= -127; power --)
    {
        ST.motor(1, power);
        ST.motor(2, power);
        delay(20);
    }
}
Sabertooth Simplified Serial Library for Arduino
Control your Sabertooth with Simplified Serial.

**TankStyleSweep.ino**

Sweeps various ranges in mixed (rover) mode.

```cpp
// Tank-Style Sweep Sample
// Copyright (c) 2012 Dimension Engineering LLC
// See license.txt for license details.

#include <SabertoothSimplified.h>

// Mixed mode is for tank-style diff-drive robots.
// Only Packet Serial actually has mixed mode, so
// this Simplified Serial library
// emulates it (to allow easy switching between the
two libraries).

SabertoothSimplified ST; // We'll name the
// Sabertooth object ST.

// For how to configure the Sabertooth, see the
// DIP Switch Wizard for

// http://www.dimensionengineering.com/datasheets/
// SabertoothDIPWizard/start.htm

// Be sure to select Simplified Serial Mode for
// use with this library.

// This sample uses a baud rate of 9600.

// Connections to make:
// Arduino TX->1   ->   Sabertooth S1
// Arduino GND    ->   Sabertooth 0V
```
// Arduino VIN -> Sabertooth 5V (OPTIONAL, if you want the Sabertooth to power the Arduino)

// If you want to use a pin other than TX->1, see the SoftwareSerial example.

void setup()
{
  SabertoothTXPinSerial.begin(9600); // This is the baud rate you chose with the DIP switches.

  ST.drive(0); // The Sabertooth won't act on mixed mode until
  ST.turn(0); // it has received power levels for BOTH throttle and turning, since it
  // mixes the two together to get diff-drive power levels for both motors.
  // So, we set both to zero initially.
}

// Mixed mode tips:
// drive() should go forward and back, turn() should go right and left.
// If this is reversed, swap M2A and M2B.
// Positive on drive() should go forward, negative should go backward.
// If this is reversed, swap A and B on both M1 and M2.
// Positive on turn() should go right, negative should go left.
// If this is reversed, swap M1 and M2.

// In this sample, the SLOW sweep (left-to-right) here is turning,
// and the FAST sweep (backwards-to-forwards) is throttle.

void loop()
{ int power;

// Don't turn. Ramp from going backwards to going forwards, waiting 20 ms (1/50th of a second) per value.
for (power = -127; power <= 127; power ++)
{
    ST.drive(power);
    delay(20);
}

// Now, let's use a power level of 20 (out of 127) forward.
// This way, our turning will have a radius.
// Mostly, the command
// is just to demonstrate you can use drive() and turn() at the same time.
ST.drive(20);

// Ramp turning from full left to full right SLOWLY by waiting 50 ms (1/20th of a second) per value.
for (power = -127; power <= 127; power ++)
{
    ST.turn(power);
    delay(50);
}

// Now stop turning, and stop driving.
ST.turn(0);
ST.drive(0);

// Wait a bit. This is so you can catch your robot if you want to. :-)
delay(5000);}

Sabertooth Simplified Serial Library for Arduino

Control your Sabertooth with Simplified Serial.

<table>
<thead>
<tr>
<th>Main Page</th>
<th>Classes</th>
<th>Files</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class List</td>
<td>Class Index</td>
<td>Class Members</td>
<td></td>
</tr>
</tbody>
</table>

**SabertoothSimplified Member List**

This is the complete list of members for **SabertoothSimplified**, including all inherited members.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>drive(int power)</code></td>
<td>SabertoothSimplified</td>
</tr>
<tr>
<td><code>motor(int power)</code></td>
<td>SabertoothSimplified</td>
</tr>
<tr>
<td><code>motor(byte motor, int power)</code></td>
<td>SabertoothSimplified</td>
</tr>
<tr>
<td><code>SabertoothSimplified()</code></td>
<td>SabertoothSimplified</td>
</tr>
<tr>
<td><code>SabertoothSimplified(Print &amp;port)</code></td>
<td>SabertoothSimplified</td>
</tr>
<tr>
<td><code>stop()</code></td>
<td>SabertoothSimplified</td>
</tr>
<tr>
<td><code>turn(int power)</code></td>
<td>SabertoothSimplified</td>
</tr>
</tbody>
</table>