Control your Sabertooth with Simplified Serial.

Class List Class Index Class Members	Main Page	е	Classes		Files	Ex	amples		
	Class List	С	Class Index C		lass Memb	ers			

Class List

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

G SabertoothSimplified	Controls a Sabertooth motor
	driver running in Simplified Serial
	mode

Control your Sabertooth with Simplified Serial.

Main Page	е	Classes	Files	Ex	amples	
Class List	С	lass Index	Class Memb	oers		
Saberto Class R	oot lef	hSimplerence	ified	Public	Member Fund	tions List of all members

Controls a Sabertooth motor driver running in Simplified Serial mode. More...

Public Member Functions

SabertoothSimplified ()

SabertoothSimplified (Print &port)

void **motor** (int power)

void **motor** (byte motor, int power)

void **drive** (int power)

void **turn** (int power)

void stop ()

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

void SabertoothSimplified::drive(int power)

Sets the driving power.

Parameters

power The power, between -127 and 127.

Examples:

TankStyleSweep.ino.

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.

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.

Main Page Classes Files Examples								
Class List Class Index Class Members								
Class In	Class Index							
S								
S								
SabertoothSimplified								
S								

Control your Sabertooth with Simplified Serial.

Maiı	n Page	e C	lasses	Files	Ex	amples	
Class List Class Index		Index	Class Memb	ers			
All	All Functions						

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

Main Page		e C	lasses	Files	Ex	amples	
Class	List	Class	Index	Class Memb	ers		
All	Func	tions					

- drive() : SabertoothSimplified
- motor() : SabertoothSimplified
- SabertoothSimplified(): SabertoothSimplified
- stop() : SabertoothSimplified
- turn() : SabertoothSimplified

Main Pa	ge	Classes	Files	Examples	
File List					
File Lis	st				
Here is a	a list	of all docu	mented f	iles with brief	descriptions:
	_				[detail level 12]

🗸 💼 SabertoothSimplified	
SabertoothSimplified.h	

Main Page Classes Files Examples								
SabertoothSimplified								
SabertoothSimplified Directory Reference								

Files

file SabertoothSimplified.cpp

file SabertoothSimplified.h [code]

Control your Sabertooth with Simplified Serial.

Main Pag	ge	Classes	Files	Examples	
File List					
SabertoothS	implif	ied >			

SabertoothSimplified.h

```
1 /*
 2 Arduino Library for Sabertooth Simplified
 Serial
 3 Copyright (c) 2012-2013 Dimension Engineering
 LLC
 4 http://www.dimensionengineering.com/arduino
 5
 6 Permission to use, copy, modify, and/or
 distribute this software for any
 7 purpose with or without fee is hereby
 granted, provided that the above
8 copyright notice and this permission notice
 appear in all copies.
 9
10 THE SOFTWARE IS PROVIDED "AS IS" AND THE
 AUTHOR DISCLAIMS ALL WARRANTIES
11 WITH REGARD TO THIS SOFTWARE INCLUDING ALL
 IMPLIED WARRANTIES OF
12 MERCHANTABILITY AND FITNESS. IN NO EVENT
 SHALL THE AUTHOR BE LIABLE FOR ANY
13 SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL
 DAMAGES OR ANY DAMAGES WHATSOEVER
14 RESULTING FROM LOSS OF USE, DATA OR PROFITS,
 WHETHER IN AN ACTION OF CONTRACT,
```

```
15 NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING
 OUT OF OR IN CONNECTION WITH THE
16 USE OR PERFORMANCE OF THIS SOFTWARE.
17 */
18
19 #ifndef SabertoothSimplified_h
20 #define SabertoothSimplified_h
21
22 #if defined(ARDUINO) && ARDUINO >= 100
23 #include <Arduino.h>
24 #else
25 #include <WProgram.h>
26 #endif
27
28 #if defined(USBCON)
29 #define SabertoothTXPinSerial Serial1 //
 Arduino Leonardo has TX->1 on Serial1, not
 Serial.
30 #else
31 #define SabertoothTXPinSerial Serial
32 #endif
33 #define SyRenTXPinSerial
 SabertoothTXPinSerial
34
39 class SabertoothSimplified
40 {
41 public:
     SabertoothSimplified();
46
47
     SabertoothSimplified(Print& port);
53
54
   public:
55
     void motor(int power);
60
61
     void motor(byte motor, int power);
67
68
73 void drive(int power);
```

```
74
     void turn(int power);
79
80
     void stop();
84
85
86
   private:
     void mixedMode(boolean enable);
87
     void mixedUpdate();
88
     void raw(byte motor, int power);
89
90
91
   private:
92
     boolean _mixed;
93
     int
            __mixedDrive, __mixedTurn;
     boolean _mixedDriveSet, _mixedTurnSet;
94
95
     Print& __port;
96 };
97
98 #endif
```

Control your Sabertooth with Simplified Serial.

Main Page	Classes	Files	Examples	
Examples	;			

Here is a list of all examples:

- SimpleExample/SimpleExample.ino
- SoftwareSerial/SoftwareSerial.ino
- Sweep/Sweep.ino
- TankStyleSweep.ino

Control your Sabertooth with Simplified Serial.

Main Page Classes Files Examples

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
 11
    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.
 11
 // 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);
}
```

Control your Sabertooth with Simplified Serial.

|--|

SoftwareSerial/SoftwareSerial.ino

Uses a pin other than TX to connect to S1.

```
// 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()
ł
 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);
}
```

Control your Sabertooth with Simplified Serial.

Main Page Classes	Files	Examples	
-------------------	-------	----------	--

Sweep/Sweep.ino

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

```
// 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
 11
    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.
 11
 // 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);
 }
}
```

Control your Sabertooth with Simplified Serial.

Main Page	Classes	Files	Examples	
-----------	---------	-------	----------	--

TankStyleSweep.ino

Sweeps various ranges in mixed (rover) mode.

```
// 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.
 11
// Connections to make:
    Arduino TX->1 -> Sabertooth S1
 11
11
    Arduino GND -> Sabertooth OV
```

```
// Arduino VIN -> Sabertooth 5V (OPTIONAL,
    if you want the Sabertooth to power the
    Arduino)
11
// 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()
11
    should go right and left.
11
       If this is reversed, swap M2A and M2B.
11
    Positive on drive() should go forward,
    negative should go backward.
11
       If this is reversed, swap A and B on both M1
    and M2.
    Positive on turn() should go right, negative
11
    should go left.
       If this is reversed, swap M1 and M2.
11
// 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);
}
```

Control your Sabertooth with Simplified Serial.

Main Page	e Classes	Files	Ex	amples	
Class List	Class Index	Class Members			

SabertoothSimplified Member List

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

drive(int power)	SabertoothSimplified
motor(int power)	SabertoothSimplified
motor(byte motor, int power)	SabertoothSimplified
SabertoothSimplified()	SabertoothSimplified
SabertoothSimplified(Print &port)	SabertoothSimplified
stop()	SabertoothSimplified
turn(int power)	SabertoothSimplified