Home

CLASSES

rcb
- endScript
- getBoardId
- getBoardVersion
- getFirmwareVersion
- onKeyboardPress
- setDebugMode
- wait

console
- append
- clear
- error
- overwrite
- print
- remove
- setVerbose
- warning

files
- appendTextFile
- newLogEntry
- newLogFile
- newTextFile

output
- ramp
- set
- steps
- steps2

sensors
- averageResultsArray
- read
- readOhm
- setMotorPoles
- setSafetyLimit
- tareCurrent
- tareLoadCells

udp
- ab2str
- init
- onReceive
- send
- str2ab

GLOBAL

output

rcb. output

`new output()`

Output control interface functions.

Source: rcbApi.js, line 457

Methods

`(static) ramp(outputId, from, to, duration, callback)`

Smoothly ramps up or down the pwm signal. For safety reasons, will only work if the output was previously activated using the rcb.output.set function. To cancel/update a ramp in progress, simply call this function again with new parameters. For example if you want to stop the ramp with the output at 1000us, call: rcb.output.ramp("esc", 1000, 1000, 0, null); Note that only one control function can be used simultaneously. You must wait for the ramp/steps function to start another.

Parameters:

Name	Type	Description
`outputId`	string	outputId - "esc", "servo1", "servo2", "servo3", "escA", "escB", "servoA", "servoB". escA, servoA, escB, servoB are for the Series 1780 coaxial channels. For the mono version of Series 1780, use the A side. Can also be an array of multiple outputs eg. ['escA','servoA'].
`from`	number	Ramp starting value between 0 and 2300 microseconds. Must be an array if the first parameter is also an array.
`to`	number	Ramp finishing value between 0 and 2300 microseconds. Must be an array if the first parameter is also an array.

Home

CLASSES

rcb
- endScript
- getBoardId
- getBoardVersion
- getFirmwareVersion
- onKeyboardPress
- setDebugMode
- wait

console
- append
- clear
- error
- overwrite
- print
- remove
- setVerbose
- warning

files
- appendTextFile
- newLogEntry
- newLogFile
- newTextFile

output
- ramp
- set
- steps
- steps2

sensors
- averageResultsArray
- read
- readOhm
- setMotorPoles
- setSafetyLimit
- tareCurrent
- tareLoadCells

udp
- ab2str
- init
- onReceive
- send
- str2ab

GLOBAL

Name	Type	Description
`duration`	number	The duration of the ramp in seconds.
`callback`	rampDone	Function to execute when the ramp is finished.

Source: rcbApi.js, line 620

Examples

```
//Illustrates the use of the ramp function
rcb.console.print("Initializing ESC...");
rcb.output.set("esc",1000);
rcb.wait(callback, 4);

function callback(){
    var from = 1000;
    var to = 1400;
    var duration = 15;
    var done = rcb.endScript;
    rcb.output.ramp("esc", from, to, duration, done);
}
```

```
//Same as example above but with multiple outputs simultaneously
rcb.console.print("Initializing ESC...");
var outputs = ["escA","escB"];
var minVal = [1000,1000];
var maxVal = [1400,1300];
rcb.output.set(outputs,minVal);
rcb.wait(callback, 4);

function callback(){
    var duration = 15;
    var done = rcb.endScript;
    rcb.output.ramp(outputs, minVal, maxVal, duration, done);
}
```

(static) set(outputId, value, protocol*opt*)

Sets the control output. The first time calling this function the protocol must be specified to activate the output.

Parameters:

Name	Type	Attributes	Description

Home

CLASSES

rcb
- endScript
- getBoardId
- getBoardVersion
- getFirmwareVersion
- onKeyboardPress
- setDebugMode
- wait

console
- append
- clear
- error
- overwrite
- print
- remove
- setVerbose
- warning

files
- appendTextFile
- newLogEntry
- newLogFile
- newTextFile

output
- ramp
- set
- steps
- steps2

sensors
- averageResultsArray
- read
- readOhm
- setMotorPoles
- setSafetyLimit
- tareCurrent
- tareLoadCells

udp
- ab2str
- init
- onReceive
- send
- str2ab

GLOBAL

Name	Type	Attributes	Description
`outputId`	string		"esc", "servo1", "servo2", "servo3", "escA", "escB", "servoA", "servoB". escA, servoA, escB, servoB are for the Series 1780 coaxial channels. For the mono version of Series 1780, use the A side. Can also be an array of multiple outputs eg. ['escA','servoA'].
`value`	number		range depends on the protocol used. Must be an array if the first parameter is also an array. Any value outside the protocol's supported range will turn off the output. See the Utilities tab to learn more about the available control protocols and their respective value ranges.
`protocol`	boolean	<optional>	If using the external RCbenchmark control board you can switch the protocol to one of these: "pwm_50", "pwm_100", "pwm_200", "pwm_300", "pwm_400", "pwm_500", "dshot150", "dshot300", "dshot600", "dshot1200", "multishot", "oneshot42", "oneshot125".

Source: rcbApi.js, line 486

Example

```
// Activate the PWM output
rcb.output.set("esc", 1000);
rcb.wait(callback1, 4);

// Setting outside the valid range (700-2300 for pwm) will turn
function callback1(){
    rcb.output.set("esc", 3000);
    rcb.wait(callback2, 4);
}

// You can switch control protocol if using the RCB control boar
// https://www.rcbenchmark.com/products/series-1580-1585-rc-con
```

Home

CLASSES

rcb
- endScript
- getBoardId
- getBoardVersion
- getFirmwareVersion
- onKeyboardPress
- setDebugMode
- wait

console
- append
- clear
- error
- overwrite
- print
- remove
- setVerbose
- warning

files
- appendTextFile
- newLogEntry
- newLogFile
- newTextFile

output
- ramp
- set
- steps
- steps2

sensors
- averageResultsArray
- read
- readOhm
- setMotorPoles
- setSafetyLimit
- tareCurrent
- tareLoadCells

udp
- ab2str
- init
- onReceive
- send
- str2ab

GLOBAL

```
    function callback2(){
        rcb.output.set("esc", 0, "dshot150");
        rcb.wait(callback3, 4);
    }

    function callback3(){
        rcb.output.set("esc", 500);
        rcb.wait(rcb.endScript, 4);
    }
```

(static) steps(outputId, from, to, steps, callback*opt*)

NOTE: Consider the steps2 function instead, which has a few extra features like a cooldown time. Steps up or down the pwm signal allowing you to perform tasks between each step. For safety reasons, will only work if the output was previously activated using the rcb.output.set function. Note that only one control function can be used simultaneously. You must wait for the steps/ramp function to finish to start another.

Parameters:

Name	Type	Attributes	Description
`outputId`	string		outputId - "esc", "servo1", "servo2", "servo3", "escA", "escB", "servoA", "servoB". escA, servoA, escB, servoB are for the Series 1780 coaxial channels. For the mono version of Series 1780, use the A side. Can also be an array of multiple outputs eg. ['escA','servoA'].
`from`	number		Steps starting value between 0 and 2300 microseconds. Must be an array if the first parameter is also an array.
`to`	number		Steps finishing value between 0 and 2300 microseconds. Must be an array if the first parameter is also an array.
`steps`	integer		Number of steps to perform.

Home

CLASSES

rcb
- endScript
- getBoardId
- getBoardVersion
- getFirmwareVersion
- onKeyboardPress
- setDebugMode
- wait

console
- append
- clear
- error
- overwrite
- print
- remove
- setVerbose
- warning

files
- appendTextFile
- newLogEntry
- newLogFile
- newTextFile

output
- ramp
- set
- steps
- steps2

sensors
- averageResultsArray
- read
- readOhm
- setMotorPoles
- setSafetyLimit
- tareCurrent
- tareLoadCells

udp
- ab2str
- init
- onReceive
- send
- str2ab

GLOBAL

Name	Type	Attributes	Description
`callback`	stepDone	<optional>	Function to execute when a step finishes. This function should introduce some sort of delay for the steps function to be effective.

Source: rcbApi.js, line 724

Example

```
//Illustrates the use of the steps function
rcb.console.print("Initializing ESC...");
rcb.output.set("esc",1000);
rcb.wait(callback, 4);
var sensorPrintId;

function callback(){
    var from = 1000;
    var to = 1400;
    var steps = 10;
    rcb.output.steps("esc", from, to, steps, stepFct);
}

//Function called at every step
function stepFct(lastStep, nextStepFct){
    if(lastStep){
        rcb.endScript();
    }else{
        rcb.console.setVerbose(false);
        rcb.wait(function(){ //2 seconds settling time

            //Do stuff here... (record to log file, calculate s
            rcb.sensors.read(readDone);

        }, 2);
    }

    //Function called when read complete
    function readDone(result){
        var speed = result.motorElectricalSpeed.displayValue;
        var unit = result.motorElectricalSpeed.displayUnit;
        if(sensorPrintId === undefined) sensorPrintId = rcb.con
        rcb.console.overwrite("Motor Speed: " + speed + " " + u

        //When done working, go to the next step
        rcb.console.setVerbose(true);
        nextStepFct();
    }
}
```

`(static) steps2(outputId, from, to, steps, stepCallback, finishedCallback, params`*opt*`)`

Home

CLASSES

rcb
- endScript
- getBoardId
- getBoardVersion
- getFirmwareVersion
- onKeyboardPress
- setDebugMode
- wait

console
- append
- clear
- error
- overwrite
- print
- remove
- setVerbose
- warning

files
- appendTextFile
- newLogEntry
- newLogFile
- newTextFile

output
- ramp
- set
- steps
- steps2

sensors
- averageResultsArray
- read
- readOhm
- setMotorPoles
- setSafetyLimit
- tareCurrent
- tareLoadCells

udp
- ab2str
- init
- onReceive
- send
- str2ab

GLOBAL

Compared to the steps function, the steps2 function has built-in cooldown time, settling time, and signal rate limiting. Steps up or down the pwm signal allowing you to perform tasks between each step. For safety reasons, will only work if the output was previously activated using the rcb.output.set function. Note that only one control function can be used simultaneously. You must wait for the steps/ramp function to finish to start another. You can set the "from" value higher than the "to" value, in which case the steps will go downwards. Important: the rate limiting is only between the steps. You should bring the throttle up to the correct throttle yourself and bring it back down yourself external to this function.

Parameters:

Name	Type	Attributes	Description
`outputId`	string		outputId - "esc", "servo1", "servo2", "servo3", "escA", "escB", "servoA", "servoB". escA, servoA, escB, servoB are for the Series 1780 coaxial channels. For the mono version of Series 1780, use the A side. Can also be an array of multiple outputs eg. ['escA','servoA'].
`from`	number		Steps starting value between 0 and 2300 microseconds. Must be an array if the first parameter is also an array.
`to`	number		Steps finishing value between 0 and 2300 microseconds. Must be an array if the first parameter is also an array.
`steps`	integer		Number of steps to perform.

Home

CLASSES

rcb
- endScript
- getBoardId
- getBoardVersion
- getFirmwareVersion
- onKeyboardPress
- setDebugMode
- wait

console
- append
- clear
- error
- overwrite
- print
- remove
- setVerbose
- warning

files
- appendTextFile
- newLogEntry
- newLogFile
- newTextFile

output
- ramp
- set
- steps
- steps2

sensors
- averageResultsArray
- read
- readOhm
- setMotorPoles
- setSafetyLimit
- tareCurrent
- tareLoadCells

udp
- ab2str
- init
- onReceive
- send
- str2ab

GLOBAL

Name	Type	Attributes	Description
`stepCallback`	stepCallback		Function to execute when a step finishes. This function should introduce some sort of delay for the steps function to be effective (by calling the nextStep function).
`finishedCallback`	finishedCallback		Function to execute when the steps are all done.

Home

CLASSES

rcb
- endScript
- getBoardId
- getBoardVersion
- getFirmwareVersion
- onKeyboardPress
- setDebugMode
- wait

console
- append
- clear
- error
- overwrite
- print
- remove
- setVerbose
- warning

files
- appendTextFile
- newLogEntry
- newLogFile
- newTextFile

output
- ramp
- set
- steps
- steps2

sensors
- averageResultsArray
- read
- readOhm
- setMotorPoles
- setSafetyLimit
- tareCurrent
- tareLoadCells

udp
- ab2str
- init
- onReceive
- send
- str2ab

GLOBAL

Name	Type	Attributes	Description
`params`	Object	<optional>	Optional parameters: {steps_qty: default=5, settlingTime_s: default=2, cooldownTime_s: default=0, cooldownThrottle_us: default=from, cooldownMinThrottle: default=0, max_slew_rate_us_per_s: default=100}. Some motors heat up quickly, and may require a cooldown period between steps. You can also optionally specify the throttle at which to cooldown. When activating the cooldown function, the time to complete a test dramatically increases. For this reason the cooldownMinThrottle setting lets the cooldown activate only at high throttle (by default all steps are followed by a cooldown step). Some motors can generate too much torque from step inputs, and some power supplies will not tolerate a motor spinning down quickly. For this reason you can set a rate limit to the throttle signal. Zero disables the rate-limit feature.

Source: rcbApi.js, line 785

Home

CLASSES

rcb
- endScript
- getBoardId
- getBoardVersion
- getFirmwareVersion
- onKeyboardPress
- setDebugMode
- wait

console
- append
- clear
- error
- overwrite
- print
- remove
- setVerbose
- warning

files
- appendTextFile
- newLogEntry
- newLogFile
- newTextFile

output
- ramp
- set
- steps
- steps2

sensors
- averageResultsArray
- read
- readOhm
- setMotorPoles
- setSafetyLimit
- tareCurrent
- tareLoadCells

udp
- ab2str
- init
- onReceive
- send
- str2ab

GLOBAL

Example

```
See the sample script called "Sweep - discrete V2" for usage sa
```

Documentation generated by JSDoc 3.6.3 on Wed Aug 21 2019 09:49:26 GMT-0400 (Eastern Daylight Time) using the Minami theme.