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

rcb

rcb

`new rcb(args)`

Api constructor. Called by the GUI to start the script (do not use in scripts).

Parameters:

Name	Type	Description
`args`	Object	GUI arguments.

Source: rcbApi.js, line 79

Classes

console

files

output

sensors

udp

Methods

`(static) endScript()`

Finishes the script execution. If this function is not called, the user will have to press the "Stop" button to stop the script.

Source: rcbApi.js, line 407

`(static) getBoardId() → {string}`

Returns the board's unique ID.

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

Source: rcbApi.js, line 384

Returns:

A string representing the board's unique serial number.

Type `string`

Example

```
var boardId = rcb.getBoardId();
rcb.console.print(boardId);
rcb.endScript();
```

`(static) getBoardVersion() → {string}`

Returns the board's hardware version.

Source: rcbApi.js, line 392

Returns:

A string representing the board's hardware version.

Type `string`

`(static) getFirmwareVersion() → {string}`

Returns the board's firmware version.

Source: rcbApi.js, line 400

Returns:

A string representing the board's firmware version.

Type `string`

`(static) onKeyboardPress(callback)`

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

Allows for interactive scripts by triggering a special callback when a key is pressed. To use the 'enter' key, make sure the focus is not on the 'stop script' button otherwise the script will stop. The callback you specify will be returned with the ASCII value of the key pressed. For example, spacebar is 32.

Parameters:

Name	Type	Description
`callback`	keyPressed	Function to execute when a key is pressed

Source: rcbApi.js, line 448

Example

```
// Example illustrating how to use the onKeyboar

rcb.console.print("Listening for keypress...");

// Setup keypress callback function
rcb.onKeyboardPress(function(key){
    // Print on screen which key was pressed
    var ascii = String.fromCharCode(key);
    rcb.console.print("You pressed " + ascii + "
});
```

(static) setDebugMode(enable)

Activates or deactivates the debug mode.

Parameters:

Name	Type	Description
`enable`	boolean	Set to "true" to activate debug mode, "false" otherwise.

Source: rcbApi.js, line 418

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

`(static) wait(callback, delay)`

Waits a certain number of seconds before executing the callback function. Note that calling this function again will cancel a previous wait. Use the javascript setTimeout function if you need multiple delays in parallel.

Parameters:

Name	Type	Description
`callback`	waitDone	The function to execute after the delay is over.
`delay`	number	Wait delay in seconds (can be floating numbers like 0.1 for 100ms).

Source: rcbApi.js, line 1191

Example

```
//Illustrates the use of the wait and overwrite
rcb.console.print("LEGEND...");
rcb.console.setVerbose(false);
rcb.wait(callback1, 2);

function callback1(){
    rcb.console.overwrite("LEGEND... wait for it
    rcb.wait(callback2, 2);
}

function callback2(){
    rcb.console.overwrite("LEGEND... wait for it
    rcb.wait(callback3, 1.5);
}

function callback3(){
    rcb.console.overwrite("LEGENDARY!");
    rcb.endScript();
}
```

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