files

rcb. files

```
new files()
```

File interface functions

Source:	rcbApi.js, line 1210

Methods

(static) appendTextFile(text, callback*opt*)

Appends new text to the created text file. Generates an error if the function newTextFile was never called.

Parameters:

Name	Type	Attributes	Description
text	String		The raw text to append to the file.
callback	textSaved	<optional>	The function to execute when recording is done.

Source:	rcbApi.js, line 1354

(static) newLogEntry(readings, callback*opt*, additionalValues*opt*)

Records a new entry to the created log file. Generates an error if the function newLogFile was never called.

Parameters:

Name	Type	Attributes	Description
readings	Object		Sensor readings as returned by the readSensors function.
callback	newLogSaved	<optional>	The function to execute when recording is done.
additionalValues	Array.<(number|string)>	<optional>	Additional values to append to the entry.

Source:	rcbApi.js, line 1298

(static) newLogFile(params*opt*)

Creates a new file for logging (in the user's working directory). Generates an error if user's working directory is not set.

Parameters:

Name	Type	Attributes	Description

Name	Type	Attributes	Description
params	Object	<optional>	parameters for file creation.

Properties

Name	Type	Attributes	Default	Description
prefix	string	<optional>	"Auto"	A prefix to append in front of the file.
additionalHeaders	Array.<string>	<optional>		Additional header(s) in addition to the standard headers.

Source: rcbApi.js, line 1278

Examples

```
//Example script recording 5 rows in a loop
var numberOflines = 5;

//Create a new log
rcb.files.newLogFile({prefix: "Example1"});

//Start the sequence
readSensor();

function readSensor(){
    if(numberOflines>0){
        rcb.sensors.read(readDone);
        numberOflines--;
    }else
        rcb.endScript();
}

function readDone(result){
    rcb.files.newLogEntry(result,readSensor);
}
```

```
//Example recording with extra data
var numberOfLines = 5;

//Create a new log
var add = ["Remaining", "Line"];
rcb.files.newLogFile({prefix: "Example2", additionalHeaders: add});

//Start the sequence
readSensor();

function readSensor(){
    if(numberOfLines>0){
        rcb.sensors.read(readDone);
        numberOfLines--;
    }else
        rcb.endScript();
}

function readDone(result){
    var add = [numberOfLines, 5-numberOfLines];
    rcb.files.newLogEntry(result,readSensor,add);
}
```

```
//This example continuously records data at full
//speed until the user stops the script

//Create a new log
rcb.files.newLogFile({prefix: "Continuous"});

//Start the sequence
readSensor();

function readSensor(){
    rcb.sensors.read(saveResult,10);
}

function saveResult(result){
    rcb.files.newLogEntry(result, readSensor);
}
```

(static) newTextFile(params*opt*)

Creates a new empty file for writing raw text (in the user's working directory). Generates an error if user's working directory is not set.

Parameters:

Name	Type	Attributes	Description
params	Object	<optional>	parameters for file creation.

Properties

Name	Type	Attributes	Default	Description
prefix	string	<optional>	"Auto"	A prefix to append in front of the file.
extension	string	<optional>	"txt"	Custom file extension.

Source: rcbApi.js, line 1334

Example

```
//Example recording to a raw text file
rcb.files.newTextFile({prefix: "RawTextExample"});
rcb.files.appendTextFile("Plain raw text\r\n", function(){
    rcb.files.appendTextFile("A new line...", function(){
        rcb.files.appendTextFile("same line...", function(){
            rcb.files.appendTextFile("more...", 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.