RobotLib

1

Generated by Doxygen 1.8.2

Sun Jan 20 2013 13:34:29

Contents

Chapter 1

RobotLib

C# software wrappers for Rovio, Roomba and NXT robots, used for teaching robotics at Lincoln School of Computer Science, UK.

The wrappers implement standard API commands for each robot together with helper classess allowing for an easy access to robot's resources (sensors, actuators, etc.).

Chapter 2

Todo List

Class Roomba.Robot

Introduce helper classes for Roomba state, odometry, motors, etc. Implement iCreate specific methods.

Class Rovio.API.Movement.ReportComponent 'ui_status' is not documented,

Use the defined enums for popular commands (e.g. Resolution)

Member Rovio.API.Network.GetIP (string type) Implement the enum input parameter.

Member Rovio.API.Network.SetDDNS (string value) Fix the input parameter according to the following format: /SetDDNS.cgi?[Enable=<true|false>][Service=<dyndns|no-ip|dnsomatic>] [&User=sUsername][&Pass=sPassword][&DomainName=sDomainName][&IP=sIPAddress] [&Proxy=sProxyServer][&ProxyPort=iProxyServerPort][&ProxyUser=sProxyUsername] [&ProxyPass=sProxyPassword][&RedirectUrl=sUrl]

Member Rovio.API.Network.SetIP (string value) Fix the input value according to the following format: /SetIP.cgi?[Interface=<eth1|wlan0>][&Enable=<true|false>][&IPWay=<manually|dhcp>][&CameraName=sName] [&IP=sIP][&Netmask=sNetmask][&Gateway=sGateway][&DNS0=sDNS0][&DNS1=sDNS1][&DNS2=sDNS2]

Member Rovio.API.Network.SetWlan (string value) Fix the input parameter according to the following format: /SetWlan.cgi?[Mode=<Managed|Ad-Hoc>][&Channel=sChannel] [&ESSID=sEssid][&WepSet=<Disable|K64|K128|ASC>] [&WepAsc=sWepAsc][&Wep64type=<Wep64HEX|Wep64ASC>][&Wep64=sWep64][&Wep128type=<Wep128HEX|Wep128ASC>][&Wep128=sWep128]

Class Rovio.API.Other Partially implemented: GetStatus command only.

Member Rovio.API.Other.StatusComponent.MonitorRect fix the return value

Member Rovio.API.Time.GetTime () Implement with DateTime input parameter.

Member Rovio.API.Time.SetLogo (string type, string position) Implement the enum input parameters.

Member Rovio.API.Time.SetTime (int seconds, int time_zone) Implement with DateTime input parameter.

Member Rovio.API.User.GetMyself (bool show_privilege) Implement the proper return value: list of strings.

Member Rovio.API.User.GetUser (bool show_privilege) Implement the proper return value: list of strings.

Chapter 3

Namespace Index

3.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 4

Hierarchical Index

4.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 5

Class Index

5.1 Class List

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

Chapter 6

Namespace Documentation

6.1 Package NXT

The NXT wrapper library. Includes the Robot class, API namespace with all commands as specified by the API specification and a set of helper classes for easy access to robot's resources.

Namespaces

- package API

Classes

- class Robot

  The main robot class.
- class Component

  Interface for different convenience classes that require access to the Robot class.
- class InPort

  Input port.
- class OutPort

  Output port.
- class DeviceInfo

  Device information.

Enumerations

- enum SonarRegister : byte {
  MeasurementUnits = 0x14, **PollInterval** = 0x40, **Mode** = 0x41, **MeasurementByte0** = 0x42,
  MeasurementByte1 = 0x43, **MeasurementByte2** = 0x44, **MeasurementByte3** = 0x45, **MeasurementByte4** = 0x46,
  MeasurementByte5 = 0x47, **MeasurementByte6** = 0x48 }

  Sonar registers.
- enum SensorType : byte {
  NoSensor = 0x00, **Switch** = 0x01, **Temperature** = 0x02, **Reflection** = 0x03,
  Angle = 0x04, **LightActive** = 0x05, **LightInactive** = 0x06, **SoundDB** = 0x07,
  SoundDBA = 0x08, **Custom** = 0x09, **LowSpeed** = 0x0A, **LowSpeed9V** = 0x0B,
  Sonar = 0x0C, **NoOfSensorTypes** = 0x0D }

  Sensor types.

- enum SensorMode : byte {
  RawMode = 0x00, **BooleanMode** = 0x20, **TransitionCntMode** = 0x40, **PeriodCounterMode** = 0x60,
  PctFullScaleMode = 0x80, **CelsiusMode** = 0xA0, **FahrenheitMode** = 0xC0, **AngleStepsMode** = 0xE0,
  SlopeMask = 0x1F, **ModeMask** = 0xE0 }

  Sensor mode.
- enum OutputMode : byte {
  MotorOn = 0x01, **Brake** = 0x02, **MotorOn_Break** = 0x03, **Regulated** = 0x04,
  MotorOn_Regulated = 0x05, **Break_Regulated** = 0x06, **MotorOn_Break_Regulated** = 0x07 }

  Output mode.
- enum RegulationMode : byte { **Idle** = 0x00, **MotorSpeed** = 0x01, **MotorSync** = 0x02 }

  Regulation mode. MotorSync enables synchronisation between several outputs: has to be set on all synchronised outputs.
- enum RunState : byte { **Idle** = 0x00, **RumpUp** = 0x10, **Running** = 0x20, **RumpDown** = 0x40 }

  Output running state.

6.1.1 Detailed Description

The NXT wrapper library. Includes the Robot class, API namespace with all commands as specified by the API specification and a set of helper classes for easy access to robot's resources. The majority of comments are based directly on the official LEGO Mindstorms NXT Communication Protocol.

6.1.2 Enumeration Type Documentation

6.1.2.1 enum NXT.OutputMode : byte

Output mode.

6.1.2.2 enum NXT.RegulationMode : byte

Regulation mode. MotorSync enables synchronisation between several outputs: has to be set on all synchronised outputs.

6.1.2.3 enum NXT.RunState : byte

Output running state.

6.1.2.4 enum NXT.SensorMode : byte

Sensor mode.

6.1.2.5 enum NXT.SensorType : byte

Sensor types.

6.1.2.6 enum NXT.SonarRegister : byte

Sonar registers.

6.2 Package NXT.API

Classes

- class API

  All commands as specified by the API document.
- class DirectCommand

  Direct commands.
- class SystemCommand

  System commands.

6.3 Package Roomba

The Roomba wrapper library. Includes the Robot class, SCI namespace with all commands as specified by the SCI specification and a set of helper classes for easy access to robot's resources.

Namespaces

- package SCI

  All commands as specified by the SCI document.

Classes

- class Robot

  The main class for communication with Roomba through a serial interface.
- class Component

  Interface for different convenience classes that require access to the Robot class.
- class Sensors

  A convenience class for accessing Roomba's sensors.
- class Leds

  A convenience class for accessing Roomba's LEDs.

Enumerations

- enum ChargingState {
  NOT_CHARGING = 0, **CHARGING_RECOVERY** = 1, **CHARGING** = 2, **TRCIKLE_CHARGING** = 3,
  WAITING = 4, **CHARGING_ERROR** = 5 }

  Charging state of the robot.
- enum BaudRate {
  BAUDRATE_300 = 0, **BAUDRATE_600** = 1, **BAUDRATE_1200** = 2, **BAUDRATE_2400** = 3,
  BAUDRATE_4800 = 4, **BAUDRATE_9600** = 5, **BAUDRATE_14400** = 6, **BAUDRATE_19200** = 7,
  BAUDRATE_28800 = 8, **BAUDRATE_38400** = 9, **BAUDRATE_57600** = 10, **BAUDRATE_115200** = 11 }

  Baud rate of the serial connection.
- enum StatusLED { **STATUSLED_OFF** = 0, **STATUSLED_RED** = 1, **STATUSLED_GREEN** = 2, **STATUSLED_AMBER** = 3 }

  Status LED colour.
- enum SensorPacket { **ALL_SENSORS** = 0, **PACKET_1** = 1, **PACKET_2** = 2, **PACKET_3** = 3 }

  Sensor packet type. A packet code value of 0 sends all of the data bytes. A value of 1 through 3 sends a subset of the sensor data.

6.3.1 Detailed Description

The Roomba wrapper library. Includes the Robot class, SCI namespace with all commands as specified by the SCI specification and a set of helper classes for easy access to robot's resources. The majority of comments are based directly on the official iRobot Roomba Serial Command Interface (SCI) Specification document. http://www.-irobot.com/images/consumer/hacker/roomba_sci_spec_manual.pdf

6.3.2 Enumeration Type Documentation

6.3.2.1 enum Roomba.BaudRate

Baud rate of the serial connection.

6.3.2.2 enum Roomba.ChargingState

Charging state of the robot.

6.3.2.3 enum Roomba.SensorPacket

Sensor packet type. A packet code value of 0 sends all of the data bytes. A value of 1 through 3 sends a subset of the sensor data.

6.3.2.4 enum Roomba.StatusLED

Status LED colour.

6.4 Package Roomba.SCI

All commands as specified by the SCI document.

Classes

- class SCI

  Implementation of all SCI commands.

6.4.1 Detailed Description

All commands as specified by the SCI document.

6.5 Package Rovio

The Rovio wrapper library. Includes the Robot class, API namespace with all commands as specified by the API specification and a set of helper classes for easy access to robot's resources.

Namespaces

- package API

  All commands as specified by the API document.

Classes

- class Robot

  The main class for communication with Rovio through a web client.
- class Component

  Interface for different convenience classes that require access to the Robot class.
- class Camera

  A convenience class for acessing camera realted functionality.
- class Odometry

  A convenience class for odometry sensor.
- class IRSensor

  A convenience class for infra red proximity sensor.
- class NavigationSensor

  A convenience class for accessing the TrueTrack navigation sensor.
- class Drive

  A convinience class for driving commands.

6.5.1 Detailed Description

The Rovio wrapper library. Includes the Robot class, API namespace with all commands as specified by the API specification and a set of helper classes for easy access to robot's resources. The majority of comments are based directly on the official API Specification for Rovio document: http://www.wowwee.com/static/support/rovio/manuals/Rovio_API_Specifications_v1.2.pdf.

6.6 Package Rovio.API

All commands as specified by the API document.

Classes

- class MovementComponent

  A base class for all movement commands.
- class Movement

  All movement commands.
- class Camera

  Camera control.
- class User

  Manage user accounts.
- class Time

  Manage time settings and time zones.
- class Network

  Network management.
- class Server

  Manage server settings.
- class Mail

  Manage email settings.
- class Other

  Manage camera settings, get audio and video streams, etc.
- class API

  Contains all API commands.

6.6.1 Detailed Description

All commands as specified by the API document.

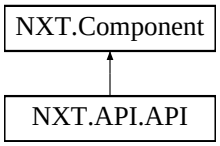
Chapter 7

Class Documentation

7.1 NXT.API.API Class Reference

All commands as specified by the API document.

Inheritance diagram for NXT.API.API:



Public Member Functions

- API (Robot _robot)

  The constructor.

Public Attributes

- DirectCommand DirectCommand

  Direct commands.
- SystemCommand SystemCommand

  System Commands

Additional Inherited Members

7.1.1 Detailed Description

All commands as specified by the API document.

7.1.2 Constructor & Destructor Documentation

7.1.2.1 NXT.API.API.API (Robot *_robot*)

The constructor.

7.1.3 Member Data Documentation

7.1.3.1 DirectCommand NXT.API.API.DirectCommand

Direct commands.

7.1.3.2 SystemCommand NXT.API.API.SystemCommand

System Commands

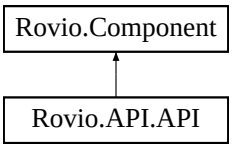
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/NXT.cs

7.2 Rovio.API.API Class Reference

Contains all API commands.

Inheritance diagram for Rovio.API.API:



Public Member Functions

- API (Robot _robot)

  The constructor.

Public Attributes

- Rovio.API.Movement Movement

  Movement commands.
- Rovio.API.Camera Camera

  Camera commands.
- Rovio.API.User User

  User commands.
- Rovio.API.Time Time

  Time commands.
- Rovio.API.Network Network

  Network commands.
- Rovio.API.Server Server

  Server commands.
- Rovio.API.Mail Mail

  Mail commands.
- Rovio.API.Other Other

  Other commands.

Additional Inherited Members

7.2.1 Detailed Description

Contains all API commands.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 Rovio.API.API.API (Robot *_robot*)

The constructor.

7.2.3 Member Data Documentation

7.2.3.1 Rovio.API.Camera Rovio.API.API.Camera

Camera commands.

7.2.3.2 Rovio.API.Mail Rovio.API.API.Mail

Mail commands.

7.2.3.3 Rovio.API.Movement Rovio.API.API.Movement

Movement commands.

7.2.3.4 Rovio.API.Network Rovio.API.API.Network

Network commands.

7.2.3.5 Rovio.API.Other Rovio.API.API.Other

Other commands.

7.2.3.6 Rovio.API.Server Rovio.API.API.Server

Server commands.

7.2.3.7 Rovio.API.Time Rovio.API.API.Time

Time commands.

7.2.3.8 Rovio.API.User Rovio.API.API.User

User commands.

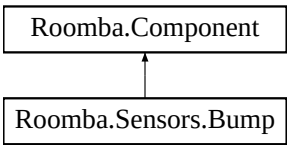
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Rovio.cs

7.3 Roomba.Sensors.Bump Class Reference

The state of the bump sensors: false = no bump, true = bump.

Inheritance diagram for Roomba.Sensors.Bump:



Public Member Functions

- Bump (Robot _robot)

  The constructor.

Properties

- bool Right [get]

  Right bump sensor.
- bool Left [get]

  Left bump sensor.

Additional Inherited Members

7.3.1 Detailed Description

The state of the bump sensors: false = no bump, true = bump.

7.3.2 Constructor & Destructor Documentation

7.3.2.1 Roomba.Sensors.Bump.Bump (Robot *_robot*)

The constructor.

7.3.3 Property Documentation

7.3.3.1 bool Roomba.Sensors.Bump.Left [get]

Left bump sensor.

7.3.3.2 bool Roomba.Sensors.Bump.Right [get]

Right bump sensor.

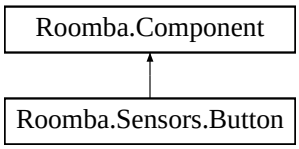
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Roomba.cs

7.4 Roomba.Sensors.Button Class Reference

The state of the four Roomba buttons: false = button not pressed, true = button pressed.

Inheritance diagram for Roomba.Sensors.Button:



Public Member Functions

- Button (Robot _robot)

  The constructor.

Properties

- bool Max [get]

  The state of the Max button.
- bool Clean [get]

  The state of the Clean button.
- bool Spot [get]

  The state of the Spot button.
- bool Power [get]

  The state of the Power button.

Additional Inherited Members

7.4.1 Detailed Description

The state of the four Roomba buttons: false = button not pressed, true = button pressed.

7.4.2 Constructor & Destructor Documentation

7.4.2.1 Roomba.Sensors.Button.Button (Robot *_robot*)

The constructor.

7.4.3 Property Documentation

7.4.3.1 bool Roomba.Sensors.Button.Clean [get]

The state of the Clean button.

7.4.3.2 bool Roomba.Sensors.Button.Max [get]

The state of the Max button.

7.4.3.3 bool Roomba.Sensors.Button.Power [get]

The state of the Power button.

7.4.3.4 bool Roomba.Sensors.Button.Spot [get]

The state of the Spot button.

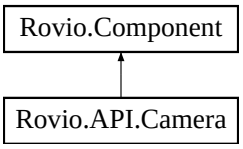
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Roomba.cs

7.5 Rovio.API.Camera Class Reference

Camera control.

Inheritance diagram for Rovio.API.Camera:



Public Types

- enum ImageResolution { QCIF = 0, CGA = 1, CIF = 2, VGA = 3 }
  Specifies the camera image resolution.
- enum ImageCompression { LOW, MED, HI }
  Specifies the camera compression ratio.
- enum CameraFlickerFrequency { AUTO = 0, F50HZ = 50, F60HZ = 60 }
  Specifies the camera anti-flickering frequency.
- enum HeadPosition { **UP**, **MIDDLE**, **DOWN** }
  Camera head position.

Public Member Functions

- Camera (Robot _robot)
  The camera class.
- Bitmap GetImage ()
  Collect a camera image (in a Bitmap format)

Properties

- ImageResolution Resolution [set]
  Set camera resolution.
- ImageCompression Compression [set]
  Set camera compression.
- int Framerate [set]
  Set image framerate.
- int Brightness [set]

Set image brightness.

- int SpeakerVolume [set]

  Set speaker volume.

- int MicVolume [set]

  Set microphone volume.

- CameraFlickerFrequency FlickerFrequency [set]

  Camera frequency.

Additional Inherited Members

7.5.1 Detailed Description

Camera control.

7.5.2 Member Enumeration Documentation

7.5.2.1 enum Rovio.API.Camera.CameraFlickerFrequency

Specifies the camera anti-flickering frequency.

Enumerator:

AUTO Auto-detect

F50HZ 50 Hz

F60HZ 60 Hz

7.5.2.2 enum Rovio.API.Camera.HeadPosition

Camera head position.

7.5.2.3 enum Rovio.API.Camera.ImageCompression

Specifies the camera compression ratio.

Enumerator:

LOW Low quality

MED Medium quality (default)

HI High quality

7.5.2.4 enum Rovio.API.Camera.ImageResolution

Specifies the camera image resolution.

Enumerator:

QCIF 176x144 pixels, QCIF resolution

CGA 320x240 pixels, CGA resolution

CIF 352x288 pixels, CIF resolution (default)

VGA 640x480 pixels, VGA resolution

7.5.3 Constructor & Destructor Documentation

7.5.3.1 Rovio.API.Camera.Camera (Robot *_robot*)

The camera class.

Parameters

_robot	

7.5.4 Member Function Documentation

7.5.4.1 Bitmap Rovio.API.Camera.GetImage ( )

Collect a camera image (in a Bitmap format)

Returns

Bitmap image

7.5.5 Property Documentation

7.5.5.1 int Rovio.API.Camera.Brightness [set]

Set image brightness.

7.5.5.2 ImageCompression Rovio.API.Camera.Compression [set]

Set camera compression.

7.5.5.3 CameraFlickerFrequency Rovio.API.Camera.FlickerFrequency [set]

Camera frequency.

7.5.5.4 int Rovio.API.Camera.Framerate [set]

Set image framerate.

7.5.5.5 int Rovio.API.Camera.MicVolume [set]

Set microphone volume.

7.5.5.6 ImageResolution Rovio.API.Camera.Resolution [set]

Set camera resolution.

7.5.5.7 int Rovio.API.Camera.SpeakerVolume [set]

Set speaker volume.

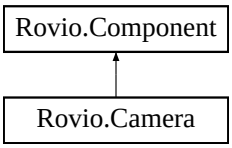
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Rovio.cs

7.6 Rovio.Camera Class Reference

A convenience class for acessing camera realted functionality.

Inheritance diagram for Rovio.Camera:



Public Member Functions

- Camera (Robot _robot)

  The constructor.
- override void Update ()

  Update all camera parameters excluding the image.
- void UpdateImage ()

  Update the camera image.

Properties

- Bitmap Image `[get]`

  The latest image from the camera.
- API.Camera.HeadPosition HeadPosition `[get, set]`

  Camera head position.
- API.Camera.ImageResolution Resolution `[get, set]`

  Image resolution.
- API.Camera.ImageCompression Compression `[get, set]`

  Image compression.
- API.Camera.CameraFlickerFrequency FlickerFrequency `[get, set]`

  Camera flicker frequency.
- int Brightness `[get, set]`

  Camera brightness.
- int Framerate `[get, set]`

  Internal frame rate.

Additional Inherited Members

7.6.1 Detailed Description

A convenience class for acessing camera realted functionality.

7.6.2 Constructor & Destructor Documentation

7.6.2.1 Rovio.Camera.Camera (Robot *_robot*)

The constructor.

Parameters

_robot	

7.6.3 Member Function Documentation

7.6.3.1 override void Rovio.Camera.Update ( ) [virtual]

Update all camera parameters excluding the image.

Reimplemented from Rovio.Component.

7.6.3.2 void Rovio.Camera.UpdateImage ( )

Update the camera image.

7.6.4 Property Documentation

7.6.4.1 int Rovio.Camera.Brightness [get], [set]

Camera brightness.

7.6.4.2 API.Camera.ImageCompression Rovio.Camera.Compression [get], [set]

Image compression.

7.6.4.3 API.Camera.CameraFlickerFrequency Rovio.Camera.FlickerFrequency [get], [set]

Camera flicker frequency.

7.6.4.4 int Rovio.Camera.Framerate [get], [set]

Internal frame rate.

7.6.4.5 API.Camera.HeadPosition Rovio.Camera.HeadPosition [get], [set]

Camera head position.

7.6.4.6 Bitmap Rovio.Camera.Image [get]

The latest image from the camera.

7.6.4.7 API.Camera.ImageResolution Rovio.Camera.Resolution [get], [set]

Image resolution.

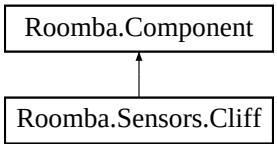
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Rovio.cs

7.7 Roomba.Sensors.Cliff Class Reference

The state of the cliff sensors: false = no cliff, true = cliff.

Inheritance diagram for Roomba.Sensors.Cliff:



Public Member Functions

- Cliff (Robot _robot)
  The constructor.

Properties

- bool Left [get]
  Left cliff sensor.
- bool FrontLeft [get]
  Front left cliff sensor.
- bool FrontRight [get]
  Front right cliff sensor.
- bool Right [get]
  Right cliff sensor.

Additional Inherited Members

7.7.1 Detailed Description

The state of the cliff sensors: false = no cliff, true = cliff.

7.7.2 Constructor & Destructor Documentation

7.7.2.1 Roomba.Sensors.Cliff.Cliff (Robot *_robot*)

The constructor.

Parameters

_robot	

7.7.3 Property Documentation

7.7.3.1 bool Roomba.Sensors.Cliff.FrontLeft [get]

Front left cliff sensor.

7.7.3.2 bool Roomba.Sensors.Cliff.FrontRight [get]

Front right cliff sensor.

7.7.3.3 bool Roomba.Sensors.Cliff.Left [get]

Left cliff sensor.

7.7.3.4 bool Roomba.Sensors.Cliff.Right [get]

Right cliff sensor.

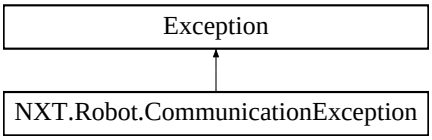
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Roomba.cs

7.8 NXT.Robot.CommunicationException Class Reference

Communication exceptions with NXT.

Inheritance diagram for NXT.Robot.CommunicationException:



Public Member Functions

- CommunicationException (int _error_code)

  The constructor.

Properties

- override string Message [get]

  The error specific message.

7.8.1 Detailed Description

Communication exceptions with NXT.

7.8.2 Constructor & Destructor Documentation

7.8.2.1 NXT.Robot.CommunicationException.CommunicationException (int *_error_code*)

The constructor.

7.8.3 Property Documentation

7.8.3.1 override string NXT.Robot.CommunicationException.Message [get]

The error specific message.

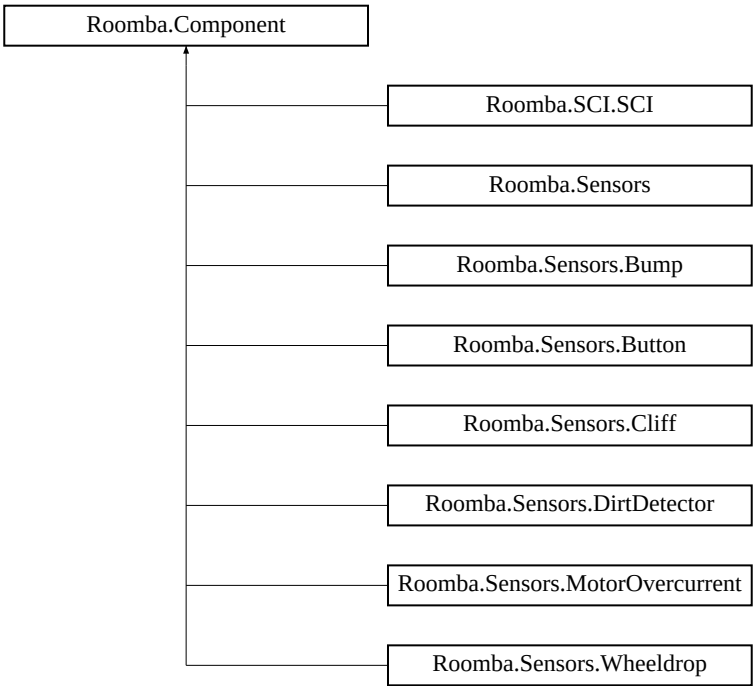
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/NXT.cs

7.9 Roomba.Component Class Reference

Interface for different convenience classes that require access to the Robot class.

Inheritance diagram for Roomba.Component:



Public Member Functions

- Component (Robot _robot)

  The constructor.

Public Attributes

- bool AutoUpdate = true

  Automatic update option for methods requesting more than one piece of information.

Protected Attributes

- Robot robot

  The robot class, accessible by all dervied classes.

7.9.1 Detailed Description

Interface for different convenience classes that require access to the Robot class.

7.9.2 Constructor & Destructor Documentation

7.9.2.1 Roomba.Component.Component (Robot *_robot*)

The constructor.

7.9.3 Member Data Documentation

7.9.3.1 bool Roomba.Component.AutoUpdate = true

Automatic update option for methods requesting more than one piece of information.

7.9.3.2 Robot Roomba.Component.robot [protected]

The robot class, accessible by all dervied classes.

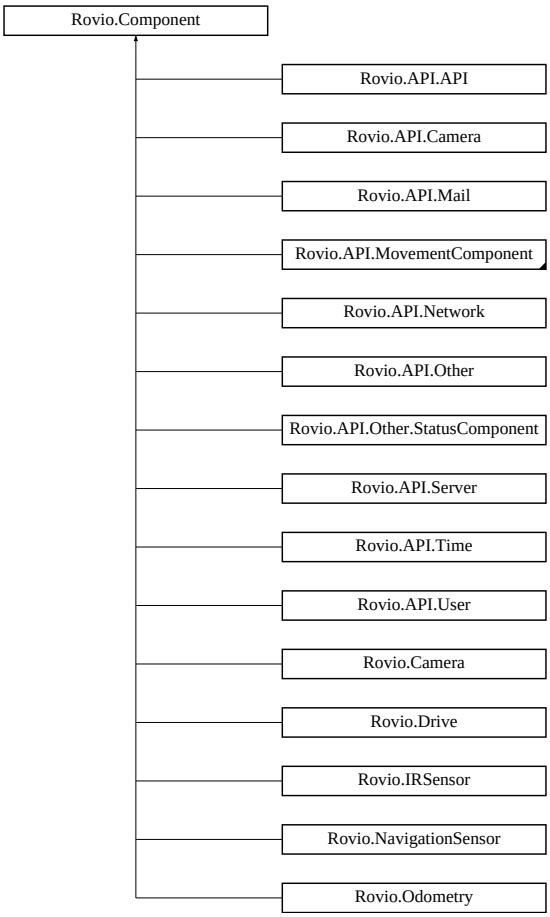
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Roomba.cs

7.10 Rovio.Component Class Reference

Interface for different convenience classes that require access to the Robot class.

Inheritance diagram for Rovio.Component:



Public Member Functions

- Component (Robot _robot)

  The constructor.
- virtual void Update ()

  The Update function that manually refreshes the state of a given component (e.g. when AutoUpdate = false)

Public Attributes

- bool AutoUpdate = true

  Automatic update option for methods requesting more than one piece of information.

Protected Attributes

- Robot robot

  The robot class, accessible by all dervied classes.

7.10.1 Detailed Description

Interface for different convenience classes that require access to the Robot class.

7.10.2 Constructor & Destructor Documentation

7.10.2.1 Rovio.Component.Component (Robot *_robot*)

The constructor.

7.10.3 Member Function Documentation

7.10.3.1 virtual void Rovio.Component.Update ( ) [virtual]

The Update function that manually refreshes the state of a given component (e.g. when AutoUpdate = false)

Reimplemented in Rovio.NavigationSensor, Rovio.IRSensor, Rovio.Odometry, Rovio.Camera, Rovio.API.Other.StatusComponent, Rovio.API.Movement.MCUReportComponent, Rovio.API.Movement.TuningParametersComponent, and Rovio.API.Movement.ReportComponent.

7.10.4 Member Data Documentation

7.10.4.1 bool Rovio.Component.AutoUpdate = true

Automatic update option for methods requesting more than one piece of information.

7.10.4.2 Robot Rovio.Component.robot [protected]

The robot class, accessible by all dervied classes.

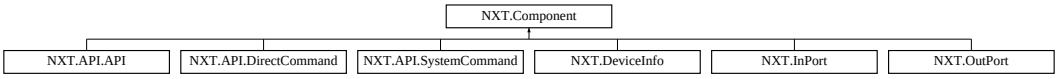
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Rovio.cs

7.11 NXT.Component Class Reference

Interface for different convenience classes that require access to the Robot class.

Inheritance diagram for NXT.Component:



Public Member Functions

- Component (Robot _robot)

  The constructor.

Public Attributes

- bool AutoUpdate = true

  Automatic update option for methods requesting more than one piece of information.

Protected Attributes

- Robot robot

  The robot class, accessible by all dervied classes.

7.11.1 Detailed Description

Interface for different convenience classes that require access to the Robot class.

7.11.2 Constructor & Destructor Documentation

7.11.2.1 NXT.Component.Component (Robot *_robot*)

The constructor.

7.11.3 Member Data Documentation

7.11.3.1 bool NXT.Component.AutoUpdate = true

Automatic update option for methods requesting more than one piece of information.

7.11.3.2 Robot NXT.Component.robot [protected]

The robot class, accessible by all dervied classes.

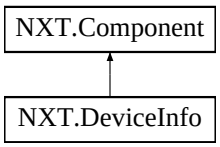
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/NXT.cs

7.12 NXT.DeviceInfo Class Reference

Device information.

Inheritance diagram for NXT.DeviceInfo:



Public Member Functions

- DeviceInfo (Robot _robot)

  The constructor.
- void Update ()

  Update the device information state.

Properties

- string Name [get, set]

  Device name
- string BTAddress [get]

  Bluetooth address.
- int BTSignalStrength [get]

  Bluetooth signal strength.
- int FreeUserFlash [get]

  Amount of free flash memory.
- int BatteryLevel [get]

  Battery level in mv.
- string ProtocolVersion [get]

  Protocol version.
- string FrimwareVersion [get]

  Firmware version.

Additional Inherited Members

7.12.1 Detailed Description

Device information.

7.12.2 Constructor & Destructor Documentation

7.12.2.1 NXT.DeviceInfo.DeviceInfo (Robot *_robot*)

The constructor.

7.12.3 Member Function Documentation

7.12.3.1 void NXT.DeviceInfo.Update ( )

Update the device information state.

7.12.4 Property Documentation

7.12.4.1 int NXT.DeviceInfo.BatteryLevel [get]

Battery level in mv.

7.12.4.2 string NXT.DeviceInfo.BTAddress [get]

Bluetooth address.

7.12.4.3 int NXT.DeviceInfo.BTSignalStrength [get]

Bluetooth signal strength.

7.12.4.4 int NXT.DeviceInfo.FreeUserFlash [get]

Amount of free flash memory.

7.12.4.5 string NXT.DeviceInfo.FrimwareVersion [get]

Firmware version.

7.12.4.6 string NXT.DeviceInfo.Name [get], [set]

Device name

7.12.4.7 string NXT.DeviceInfo.ProtocolVersion [get]

Protocol version.

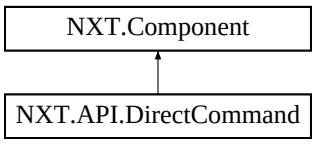
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/NXT.cs

7.13 NXT.API.DirectCommand Class Reference

Direct commands.

Inheritance diagram for NXT.API.DirectCommand:



Public Member Functions

- DirectCommand (Robot _robot)

  The constructor.
- void StartProgram (string name)

  Start the program stored in the NXT.
- void StopProgram ()

  Stop executing the current program.
- void PlaySoundFile (string name, bool loop)

  Play a specified sound file. The correct extension is .rso.
- void PlayTone (int frequency, int duration)

  Play tone with the specified frequency and duration.
- void SetOutputState (int port, int power_setpoint, OutputMode mode, RegulationMode regulation_mode, int turn_ratio, RunState run_state, int tacho_limit)

  Set output state for a specfic port.
- void SetInputMode (int port, int sensor_type, int sensor_mode)

  Set input mode for the specified port.
- void GetOutputState (int port, out byte[] response)

  Get readings from the specified output port.

- void GetInputValues (int port, out byte[] response)

  Get readings from the input port.
- void ResetInputScaledValue (int port)

  Reset the scaled value for the specified input port.
- void MessageWrite (string data, int box)

  Write a text message to the specified mailbox.
- void ResetMotorPosition (int port, bool relative)

  Reset motor position.
- int GetBatteryLevel ()

  Get battery level in mV.
- void StopSoundPlayback ()

  Stop playing the current sound.
- int KeepAlive ()

  Keep alive, return the current time limit in ms.
- int LSGetStatus (int port)

  Return the available bytes to read
- void LSWrite (int port, byte[] tx_data, int rx_data_length)

  LS Write command. tx_data should not be longer than 16 bytes.
- void LSRead (int port, out byte[] response)

  LS Read command
- string GetCurrentProgramName ()

  Get name of the current program.
- string MessageRead (int remote_inbox, int local_inbox, bool remove)

  Read message from the specified inbox.

Public Attributes

- bool RequestResponse = false

  CheckResponse will require a confirmation from NXT. It might be slower and return exceptions.

Additional Inherited Members

7.13.1 Detailed Description

Direct commands.

7.13.2 Constructor & Destructor Documentation

7.13.2.1 NXT.API.DirectCommand.DirectCommand (Robot *_robot*)

The constructor.

7.13.3 Member Function Documentation

7.13.3.1 int NXT.API.DirectCommand.GetBatteryLevel ( )

Get battery level in mV.

7.13.3.2 string NXT.API.DirectCommand.GetCurrentProgramName ( )

Get name of the current program.

7.13.3.3 void NXT.API.DirectCommand.GetInputValues (int *port,* out byte[] *response*)

Get readings from the input port.

7.13.3.4 void NXT.API.DirectCommand.GetOutputState (int *port,* out byte[] *response*)

Get readings from the specified output port.

7.13.3.5 int NXT.API.DirectCommand.KeepAlive ( )

Keep alive, return the current time limit in ms.

7.13.3.6 int NXT.API.DirectCommand.LSGetStatus (int *port*)

Return the available bytes to read

7.13.3.7 void NXT.API.DirectCommand.LSRead (int *port,* out byte[] *response*)

LS Read command

7.13.3.8 void NXT.API.DirectCommand.LSWrite (int *port,* byte[] *tx_data,* int *rx_data_length*)

LS Write command. tx_data should not be longer than 16 bytes.

7.13.3.9 string NXT.API.DirectCommand.MessageRead (int *remote_inbox,* int *local_inbox,* bool *remove*)

Read message from the specified inbox.

7.13.3.10 void NXT.API.DirectCommand.MessageWrite (string *data,* int *box*)

Write a text message to the specified mailbox.

7.13.3.11 void NXT.API.DirectCommand.PlaySoundFile (string *name,* bool *loop*)

Play a specified sound file. The correct extension is .rso.

7.13.3.12 void NXT.API.DirectCommand.PlayTone (int *frequency,* int *duration*)

Play tone with the specified frequency and duration.

Parameters

frequency	200-14000 Hz
duration	in ms

7.13.3.13 void NXT.API.DirectCommand.ResetInputScaledValue (int *port*)

Reset the scaled value for the specified input port.

7.13.3.14 void NXT.API.DirectCommand.ResetMotorPosition (int *port*, bool *relative*)

Reset motor position.

7.13.3.15 void NXT.API.DirectCommand.SetInputMode (int *port*, int *sensor_type*, int *sensor_mode*)

Set input mode for the specified port.

7.13.3.16 void NXT.API.DirectCommand.SetOutputState (int *port*, int *power_setpoint*, OutputMode *mode*, RegulationMode *regulation_mode*, int *turn_ratio*, RunState *run_state*, int *tacho_limit*)

Set output state for a specfic port.

7.13.3.17 void NXT.API.DirectCommand.StartProgram (string *name*)

Start the program stored in the NXT.

Recognised extensions: .rxe - user defined programs .rtm - try me programs .rfw - firmware .rxe - user defined programs .rpg - on-brick programs .rtm - try-me programs

7.13.3.18 void NXT.API.DirectCommand.StopProgram ( )

Stop executing the current program.

7.13.3.19 void NXT.API.DirectCommand.StopSoundPlayback ( )

Stop playing the current sound.

7.13.4 Member Data Documentation

7.13.4.1 bool NXT.API.DirectCommand.RequestResponse = false

CheckResponse will require a confirmation from NXT. It might be slower and return exceptions.

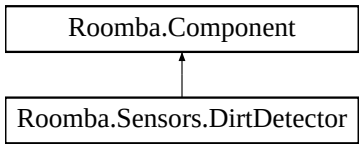
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/NXT.cs

7.14 Roomba.Sensors.DirtDetector Class Reference

The current dirt detection level (0-255) of the dirt detector. Higher values indicate higher levels of dirt detected.

Inheritance diagram for Roomba.Sensors.DirtDetector:



Public Member Functions

- DirtDetector (Robot _robot)

  The constructor.

Properties

- int Left [get]

  The current dirt level of the left dirt detector.
- int Right [get]

  The current dirt level of the right dirt detector.

Additional Inherited Members

7.14.1 Detailed Description

The current dirt detection level (0-255) of the dirt detector. Higher values indicate higher levels of dirt detected.

7.14.2 Constructor & Destructor Documentation

7.14.2.1 Roomba.Sensors.DirtDetector.DirtDetector (Robot *_robot*)

The constructor.

7.14.3 Property Documentation

7.14.3.1 int Roomba.Sensors.DirtDetector.Left [get]

The current dirt level of the left dirt detector.

7.14.3.2 int Roomba.Sensors.DirtDetector.Right [get]

The current dirt level of the right dirt detector.

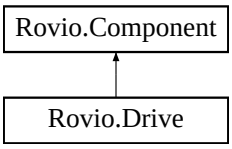
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Roomba.cs

7.15 Rovio.Drive Class Reference

A convinience class for driving commands.

Inheritance diagram for Rovio.Drive:



Public Member Functions

- Drive (Robot _robot)

  The constructor.
- void Stop ()

  Stop the robot.
- void Forward (int speed)

  Move forward.
- void Backward (int speed)

  Move backward.
- void StraightLeft (int speed)

  Move straight left.
- void StraightRight (int speed)

  Move straight right.
- void RotateLeft (int speed)

  Rotate left.
- void RotateRight (int speed)

  Rotate right.
- void DiagForwardLeft (int speed)

  Diagonal forward left.
- void DiagForwardRight (int speed)

  Diagonal forward right.
- void DiagBackwardLeft (int speed)

  Diagonal backward left.
- void DiagBackwardRight (int speed)

  Diagonal backward right.
- void RotateLeft20 (int speed)

  Rotate left by 20 degree angle increments.
- void RotateRight20 (int speed)

  Rotate right by 20 degree angle increments.

Additional Inherited Members

7.15.1 Detailed Description

A convinience class for driving commands.

7.15.2 Constructor & Destructor Documentation

7.15.2.1 Rovio.Drive.Drive (Robot *_robot*)

The constructor.

7.15.3 Member Function Documentation

7.15.3.1 void Rovio.Drive.Backward (int *speed*)

Move backward.

Parameters

speed	

7.15.3.2 void Rovio.Drive.DiagBackwardLeft (int *speed*)

Diagonal backward left.

Parameters

speed	

7.15.3.3 void Rovio.Drive.DiagBackwardRight (int *speed*)

Diagonal backward right.

Parameters

speed	

7.15.3.4 void Rovio.Drive.DiagForwardLeft (int *speed*)

Diagonal forward left.

Parameters

speed	

7.15.3.5 void Rovio.Drive.DiagForwardRight (int *speed*)

Diagonal forward right.

Parameters

speed	

7.15.3.6 void Rovio.Drive.Forward (int *speed*)

Move forward.

Parameters

speed	1 (fastest) - 10 (slowest)

7.15.3.7 void Rovio.Drive.RotateLeft (int *speed*)

Rotate left.

Parameters

speed	

7.15.3.8 void Rovio.Drive.RotateLeft20 (int *speed*)

Rotate left by 20 degree angle increments.

Parameters

speed	

7.15.3.9 void Rovio.Drive.RotateRight (int *speed*)

Rotate right.

Parameters

speed	

7.15.3.10 void Rovio.Drive.RotateRight20 (int *speed*)

Rotate right by 20 degree angle increments.

Parameters

speed	

7.15.3.11 void Rovio.Drive.Stop ( )

Stop the robot.

7.15.3.12 void Rovio.Drive.StraightLeft (int *speed*)

Move straight left.

Parameters

speed	

7.15.3.13 void Rovio.Drive.StraightRight (int *speed*)

Move straight right.

Parameters

speed	

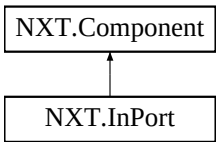
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Rovio.cs

7.16 NXT.InPort Class Reference

Input port.

Inheritance diagram for NXT.InPort:



Public Member Functions

- InPort (Robot _robot)
  The constructor.
- void Update ()
  Update the sensor state.
- void ResetScaledValue ()
  Reset the scaled value.

Public Attributes

- int Number = 0
  Port number.

Properties

- SensorMode SensorMode `[get, set]`
  Sensor mode.
- SensorType SensorType `[get, set]`
  Sensor type.
- int RawADValue `[get]`
  Raw reading value.
- int ScaledADValue `[get]`
  Scaled reading value.
- short ScaledValue `[get]`
  Scaled sensor value.
- short CalibratedValue `[get]`
  Calibrated value.

Additional Inherited Members

7.16.1 Detailed Description

Input port.

7.16.2 Constructor & Destructor Documentation

7.16.2.1 NXT.InPort.InPort (Robot *_robot*)

The constructor.

7.16.3 Member Function Documentation

7.16.3.1 void NXT.InPort.ResetScaledValue ( )

Reset the scaled value.

7.16.3.2 void NXT.InPort.Update ( )

Update the sensor state.

7.16.4 Member Data Documentation

7.16.4.1 int NXT.InPort.Number = 0

Port number.

7.16.5 Property Documentation

7.16.5.1 short NXT.InPort.CalibratedValue [get]

Calibrated value.

7.16.5.2 int NXT.InPort.RawADValue [get]

Raw reading value.

7.16.5.3 int NXT.InPort.ScaledADValue [get]

Scaled reading value.

7.16.5.4 short NXT.InPort.ScaledValue [get]

Scaled sensor value.

7.16.5.5 SensorMode NXT.InPort.SensorMode [get], [set]

Sensor mode.

7.16.5.6 SensorType NXT.InPort.SensorType [get], [set]

Sensor type.

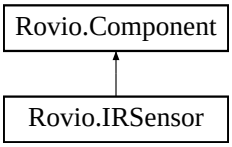
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/NXT.cs

7.17 Rovio.IRSensor Class Reference

A convenience class for infra red proximity sensor.

Inheritance diagram for Rovio.IRSensor:



Public Member Functions

- IRSensor (Robot _robot)

  The constructor.
- override void Update ()

  Update the IRSensor value.

Properties

- bool PowerOn `[get, set]`

  Activate the sensor.
- bool Detection `[get]`

  Sensor output: false - no obstacle, true - obstacle in front.

Additional Inherited Members

7.17.1 Detailed Description

A convenience class for infra red proximity sensor.

7.17.2 Constructor & Destructor Documentation

7.17.2.1 Rovio.IRSensor.IRSensor (Robot *_robot*)

The constructor.

7.17.3 Member Function Documentation

7.17.3.1 override void Rovio.IRSensor.Update ( ) `[virtual]`

Update the IRSensor value.

Reimplemented from Rovio.Component.

7.17.4 Property Documentation

7.17.4.1 bool Rovio.IRSensor.Detection `[get]`

Sensor output: false - no obstacle, true - obstacle in front.

7.17.4.2 bool Rovio.IRSensor.PowerOn [get],[set]

Activate the sensor.

The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Rovio.cs

7.18 Roomba.Leds Class Reference

A convenience class for accessing Roomba's LEDs.

Classes

- class PowerLed

  Power Led - controls the colour and intensity. iCreate left led.

Public Member Functions

- Leds (Robot _robot)

  The constructor.

Public Attributes

- PowerLed Power

  Power LED.

Properties

- bool DirtDetect [get, set]

  DirtDetect Led - blue light.
- bool Max [get, set]

  Max Led - green light. iCreate middle led.
- bool Clean [get, set]

  Clean Led - green light.
- bool Spot [get, set]

  Spot Led - green light.
- StatusLED Status [get, set]

  Status Led - controls the status led colour.

7.18.1 Detailed Description

A convenience class for accessing Roomba's LEDs.

7.18.2 Constructor & Destructor Documentation

7.18.2.1 Roomba.Leds.Leds (Robot *_robot*)

The constructor.

7.18.3 Member Data Documentation

7.18.3.1 PowerLed Roomba.Leds.Power

Power LED.

7.18.4 Property Documentation

7.18.4.1 bool Roomba.Leds.Clean [get], [set]

Clean Led - green light.

7.18.4.2 bool Roomba.Leds.DirtDetect [get], [set]

DirtDetect Led - blue light.

7.18.4.3 bool Roomba.Leds.Max [get], [set]

Max Led - green light. iCreate middle led.

7.18.4.4 bool Roomba.Leds.Spot [get], [set]

Spot Led - green light.

7.18.4.5 StatusLED Roomba.Leds.Status [get], [set]

Status Led - controls the status led colour.

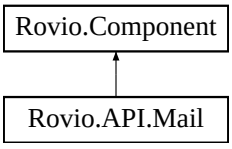
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Roomba.cs

7.19 Rovio.API.Mail Class Reference

Manage email settings.

Inheritance diagram for Rovio.API.Mail:



Public Member Functions

- Mail (Robot _robot)

  The constructor.
- void SetMail (string value)

  Configure email for sending IPCam images.
- string GetMail ()

Get email settings.

- void SendMail ()

  Send an email with IPCam images.

Additional Inherited Members

7.19.1 Detailed Description

Manage email settings.

7.19.2 Constructor & Destructor Documentation

7.19.2.1 Rovio.API.Mail.Mail (Robot *_robot*)

The constructor.

7.19.3 Member Function Documentation

7.19.3.1 string Rovio.API.Mail.GetMail ( )

Get email settings.

MailServer, Port, Sender, Receiver, Subject, Body, User, PassWord, CheckFlag and Enable

7.19.3.2 void Rovio.API.Mail.SendMail ( )

Send an email with IPCam images.

7.19.3.3 void Rovio.API.Mail.SetMail (string *value*)

Configure email for sending IPCam images.

Enable Ignored MailServer - mail server address Sender - senders email address Receiver - receivers email address, multi-receivers separated by ; Subject - subject of email User - user name for logging into the MailServer PassWord - password for logging into the MailServer CheckFlag - whether the MailServer needs to check password Interval - Ignored

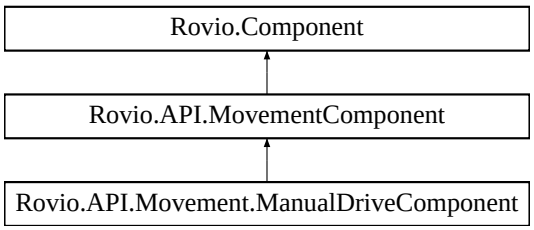
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Rovio.cs

7.20 Rovio.API.Movement.ManualDriveComponent Class Reference

Manual drive commands. Majority of the drive commands feature the speed parameter: 1 (fastest) - 10 (slowest). Note that depending on the type of surface, the robot might have problems executing commands with very low speeds (i.e. it will stall).

Inheritance diagram for Rovio.API.Movement.ManualDriveComponent:



Public Member Functions

- ManualDriveComponent (Robot _robot)
  The constructor.
- void Stop ()
  Stop the robot.
- void Forward (int speed)
  Move forward.
- void Backward (int speed)
  Move backward.
- void StraightLeft (int speed)
  Move straight left.
- void StraightRight (int speed)
  Move straight right.
- void RotateLeft (int speed)
  Rotate left.
- void RotateRight (int speed)
  Rotate right.
- void DiagForwardLeft (int speed)
  Diagonal forward left.
- void DiagForwardRight (int speed)
  Diagonal forward right.
- void DiagBackwardLeft (int speed)
  Diagonal backward left.
- void DiagBackwardRight (int speed)
  Diagonal backward right.
- void HeadUp ()
  Camera head up position.
- void HeadDown ()
  Camera head down position.
- void HeadMiddle ()
  Camera head middle position.
- void RotateLeft20 (int speed)
  Rotate left by 20 degree angle increments.
- void RotateRight20 (int speed)
  Rotate right by 20 degree angle increments.

Additional Inherited Members

7.20.1 Detailed Description

Manual drive commands. Majority of the drive commands feature the speed parameter: 1 (fastest) - 10 (slowest). Note that depending on the type of surface, the robot might have problems executing commands with very low speeds (i.e. it will stall).

7.20.2 Constructor & Destructor Documentation

7.20.2.1 Rovio.API.Movement.ManualDriveComponent.ManualDriveComponent (Robot *_robot*)

The constructor.

7.20.3 Member Function Documentation

7.20.3.1 void Rovio.API.Movement.ManualDriveComponent.Backward (int *speed*)

Move backward.

Parameters

speed	

7.20.3.2 void Rovio.API.Movement.ManualDriveComponent.DiagBackwardLeft (int *speed*)

Diagonal backward left.

Parameters

speed	

7.20.3.3 void Rovio.API.Movement.ManualDriveComponent.DiagBackwardRight (int *speed*)

Diagonal backward right.

Parameters

speed	

7.20.3.4 void Rovio.API.Movement.ManualDriveComponent.DiagForwardLeft (int *speed*)

Diagonal forward left.

Parameters

speed	

7.20.3.5 void Rovio.API.Movement.ManualDriveComponent.DiagForwardRight (int *speed*)

Diagonal forward right.

Parameters

speed	

7.20.3.6 void Rovio.API.Movement.ManualDriveComponent.Forward (int *speed*)

Move forward.

Parameters

speed	1 (fastest) - 10 (slowest)

7.20.3.7 void Rovio.API.Movement.ManualDriveComponent.HeadDown ( )

Camera head down position.

7.20.3.8 void Rovio.API.Movement.ManualDriveComponent.HeadMiddle ( )

Camera head middle position.

7.20.3.9 void Rovio.API.Movement.ManualDriveComponent.HeadUp ( )

Camera head up position.

7.20.3.10 void Rovio.API.Movement.ManualDriveComponent.RotateLeft (int *speed*)

Rotate left.

Parameters

speed	

7.20.3.11 void Rovio.API.Movement.ManualDriveComponent.RotateLeft20 (int *speed*)

Rotate left by 20 degree angle increments.

Parameters

speed	

7.20.3.12 void Rovio.API.Movement.ManualDriveComponent.RotateRight (int *speed*)

Rotate right.

Parameters

speed	

7.20.3.13 void Rovio.API.Movement.ManualDriveComponent.RotateRight20 (int *speed*)

Rotate right by 20 degree angle increments.

Parameters

speed	

7.20.3.14 void Rovio.API.Movement.ManualDriveComponent.Stop ( )

Stop the robot.

7.20.3.15 void Rovio.API.Movement.ManualDriveComponent.StraightLeft (int *speed*)

Move straight left.

Parameters

speed	

7.20.3.16 void Rovio.API.Movement.ManualDriveComponent.StraightRight (int *speed*)

Move straight right.

Parameters

speed	

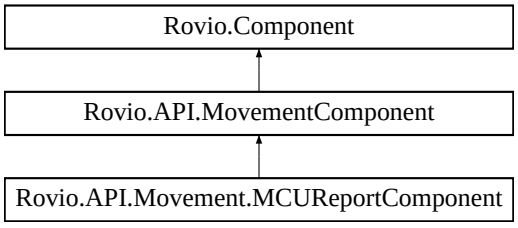
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Rovio.cs

7.21 Rovio.API.Movement.MCUReportComponent Class Reference

Provides a run-time report from Rovio's microcontroller. Run the Update() method before accessing individual parameters of the report. This solution reduces the data traffic when accessing multiple parameters at the same time.

Inheritance diagram for Rovio.API.Movement.MCUReportComponent:



Public Member Functions

- MCUReportComponent (Robot _robot)

  The constructor.
- override void Update ()

  Update the report.

Properties

- bool LeftWheelRot `[get]`

  Left wheel odometry: positive rotation direction.
- int LeftWheelTicks `[get]`

  Left wheel odometry: number of ticks.
- bool RightWheelRot `[get]`

  Right wheel odometry: positive rotation direction.
- int RightWheelTicks `[get]`

Right wheel odometry: number of ticks.

- bool RearWheelRot [get]

  Rear wheel odometry: positive rotation direction.
- int RearWheelTicks [get]

  Rear wheel odometry: number of ticks.
- int HeadPosition [get]

  Position of the head.
- int BatteryLevel [get]

  Battery level.
- int Status [get]

  Status.
- bool LightOn [get]

  The state of the front LED light.
- bool IRPowerOn [get]

  The power state of the IR proximity sensor.
- bool IRDetectorOn [get]

  The state of the IR proximity sensor.
- Movement.ChargerStatus ChargerStatus [get]

  Charger status.

Additional Inherited Members

7.21.1 Detailed Description

Provides a run-time report from Rovio's microcontroller. Run the Update() method before accessing individual parameters of the report. This solution reduces the data traffic when accessing multiple parameters at the same time.

7.21.2 Constructor & Destructor Documentation

7.21.2.1 Rovio.API.Movement.MCUReportComponent.MCUReportComponent (Robot *_robot*)

The constructor.

7.21.3 Member Function Documentation

7.21.3.1 override void Rovio.API.Movement.MCUReportComponent.Update ( ) [virtual]

Update the report.

Reimplemented from Rovio.Component.

7.21.4 Property Documentation

7.21.4.1 int Rovio.API.Movement.MCUReportComponent.BatteryLevel [get]

Battery level.

7.21.4.2 Movement.ChargerStatus Rovio.API.Movement.MCUReportComponent.ChargerStatus [get]

Charger status.

7.21.4.3 int Rovio.API.Movement.MCUReportComponent.HeadPosition [get]

Position of the head.

7.21.4.4 bool Rovio.API.Movement.MCUReportComponent.IRDetectorOn [get]

The state of the IR proximity sensor.

7.21.4.5 bool Rovio.API.Movement.MCUReportComponent.IRPowerOn [get]

The power state of the IR proximity sensor.

7.21.4.6 bool Rovio.API.Movement.MCUReportComponent.LeftWheelRot [get]

Left wheel odometry: positive rotation direction.

7.21.4.7 int Rovio.API.Movement.MCUReportComponent.LeftWheelTicks [get]

Left wheel odometry: number of ticks.

7.21.4.8 bool Rovio.API.Movement.MCUReportComponent.LightOn [get]

The state of the front LED light.

7.21.4.9 bool Rovio.API.Movement.MCUReportComponent.RearWheelRot [get]

Rear wheel odometry: positive rotation direction.

7.21.4.10 int Rovio.API.Movement.MCUReportComponent.RearWheelTicks [get]

Rear wheel odometry: number of ticks.

7.21.4.11 bool Rovio.API.Movement.MCUReportComponent.RightWheelRot [get]

Right wheel odometry: positive rotation direction.

7.21.4.12 int Rovio.API.Movement.MCUReportComponent.RightWheelTicks [get]

Right wheel odometry: number of ticks.

7.21.4.13 int Rovio.API.Movement.MCUReportComponent.Status [get]

Status.

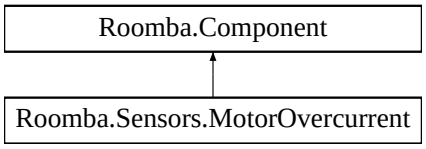
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Rovio.cs

7.22 Roomba.Sensors.MotorOvercurrent Class Reference

The state of the five motors' overcurrent sensors: false = no overcurrent, true = overcurrent.

Inheritance diagram for Roomba.Sensors.MotorOvercurrent:



Public Member Functions

- MotorOvercurrent (Robot _robot)

  The constructor.

Properties

- bool SideBrush [get]

  The state of the side brush motor overrcurrent.
- bool Vacuum [get]

  The state of the vacuum motor overrcurrent.
- bool MainBrush [get]

  The state of the main brush motor overrcurrent.
- bool DriveRight [get]

  The state of the drive right motor overrcurrent.
- bool DriveLeft [get]

  The state of the drive left motor overrcurrent.

Additional Inherited Members

7.22.1 Detailed Description

The state of the five motors' overcurrent sensors: false = no overcurrent, true = overcurrent.

7.22.2 Constructor & Destructor Documentation

7.22.2.1 Roomba.Sensors.MotorOvercurrent.MotorOvercurrent (Robot *_robot*)

The constructor.

Parameters

_robot	

7.22.3 Property Documentation

7.22.3.1 bool Roomba.Sensors.MotorOvercurrent.DriveLeft [get]

The state of the drive left motor overrcurrent.

7.22.3.2 bool Roomba.Sensors.MotorOvercurrent.DriveRight [get]

The state of the drive right motor overrcurrent.

7.22.3.3 bool Roomba.Sensors.MotorOvercurrent.MainBrush [get]

The state of the main brush motor overrcurrent.

7.22.3.4 bool Roomba.Sensors.MotorOvercurrent.SideBrush [get]

The state of the side brush motor overrcurrent.

7.22.3.5 bool Roomba.Sensors.MotorOvercurrent.Vacuum [get]

The state of the vacuum motor overrcurrent.

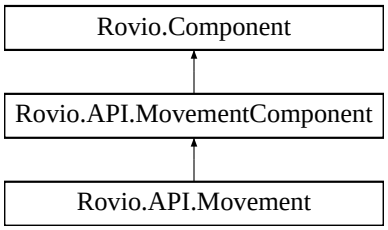
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Roomba.cs

7.23 Rovio.API.Movement Class Reference

All movement commands.

Inheritance diagram for Rovio.API.Movement:



Classes

- class ManualDriveComponent

  Manual drive commands. Majority of the drive commands feature the speed parameter: 1 (fastest) - 10 (slowest). Note that depending on the type of surface, the robot might have problems executing commands with very low speeds (i.e. it will stall).

- class MCUReportComponent

  Provides a run-time report from Rovio's microcontroller. Run the Update() method before accessing individual parameters of the report. This solution reduces the data traffic when accessing multiple parameters at the same time.

- class MovementResponseException

  Report errors in movement responses received from Rovio.

- class ReportComponent
- class TuningParametersComponent

  Manage robot parameters used during navigation: homing, docking and automatic driving.

Public Types

- enum NavigationState {
  IDLE = 0, HOMING = 1, DOCKING = 2, PLAYING_PATH = 3,
  RECORDING_PATH = 4 }

  Rovio's navigation state.
- enum ChargerStatus { **IDLE** = 0, **COMPLETED** = 1, **CHARGING** = 2, **ERROR** = 3 }

  Charger status.

Public Member Functions

- Movement (Robot _robot)

  The constructor.
- string GetReport ()

  Get the current status of the robot (in a string format). Refer to Report for accessing individual items of this report.
- void StartRecording ()

  Start recording a path.
- void AbortRecording ()

  Stop recording and discard a path.
- void StopRecording (string path)

  Stop recording and store a path.
- void DeletePath (string path)

  Delete the specified path.
- string GetPathList ()

  Get a list of stored paths.
- void PlayPathForward ()

  Replay the specified path from the closest point to the last one.
- void PlayPathBackward ()

  Replay the specified path from the closest point to the first one.
- void StopPlaying ()

  Stop playing the current path.
- void PausePlaying ()

  Pause playing the current path.
- void RenamePath (string old_path, string new_path)

  Rename the specified path.
- void GoHome ()

  Drive to home location without docking.
- void GoHomeAndDock ()

  Drive to home location with docking.
- void UpdateHomePosition ()

  Use the current location as home location.
- void SetTuningParameters ()

  Set homing, docking and driving parameters. Refer to TuningParameters for setting the individual elements.
- string GetTuningParameters ()

  Get homing, docking and driving parameters (in a string format). Refer to TuningParameters for accessing individual elements of the report.
- void ResetNavStateMachine ()

  Stop and reset to idle.
- void FrontLight (bool value)

  Switch on/off the front LED.
- void IRSensor (bool value)

Switch on/off the power of the IR sensor.

- string GetMCUReport ()

  Get the report from the robot's microcontroller. Refer to MCUReport for accessing individual items of the report.
- void ClearAllPaths ()

  Delete all stored paths.
- NavigationState GetStatus ()

  Return the navigation status.
- void SaveParameter (int index, int value)

  Store value for the specified parameter.
- int ReadParameter (int index)

  Read the specified parameter value.
- string GetLibNSVersion ()

  NorthStar/TrueTrack version.
- void EmailImage (string email_address)

  Email the current image / set an action (in path recording mode).
- void ResetHomeLocation ()

  Clear home location.

Public Attributes

- ReportComponent Report

  Report commands.
- ManualDriveComponent ManualDrive

  Manual Drive commands.
- TuningParametersComponent TuningParameters

  Tuning Parameters commands.
- MCUReportComponent MCUReport

  MCU Report commands.

Additional Inherited Members

7.23.1 Detailed Description

All movement commands.

Some of the commands perform multiple actions or return multiple data items (i.e. GetReport(), ManualDrive, GetTuningParameters(), GetMCUReport()). There is a dedicated class implemented for each such command.

Path related commands use the TrueTrack sensor and might require careful timing if issued in a sequence.

7.23.2 Member Enumeration Documentation

7.23.2.1 enum Rovio.API.Movement.ChargerStatus

Charger status.

7.23.2.2 enum Rovio.API.Movement.NavigationState

Rovio's navigation state.

Enumerator:

IDLE Idle.

HOMING Driving home.

DOCKING Docking.

PLAYING_PATH Playing a path.

RECORDING_PATH Recoring a path.

7.23.3 Member Function Documentation

7.23.3.1 string Rovio.API.Movement.GetReport ( )

Get the current status of the robot (in a string format). Refer to Report for accessing individual items of this report.

Remarks:

- Rovio will resist going outside NorthStar coverage area while recording path
- Rovio will stop recording if coverage is lost
- Rovio will stop recording if user connection is lost

7.23.3.2 string Rovio.API.Movement.GetTuningParameters ( )

Get homing, docking and driving parameters (in a string format). Refer to TuningParameters for accessing individual elements of the report.

The parameters include: LeftRight, Forward, Reverse, DriveTurn, HomingTurn, ManDrive, ManTurn and DockTimeout.

7.23.3.3 void Rovio.API.Movement.PausePlaying ( )

Pause playing the current path.

The playback continues whith the next pause command and stops completely with the stop command.

7.23.3.4 void Rovio.API.Movement.PlayPathBackward ( )

Replay the specified path from the closest point to the first one.

If the NorthStar signal is lost the playback is interrupted.

7.23.3.5 void Rovio.API.Movement.PlayPathForward ( )

Replay the specified path from the closest point to the last one.

If the NorthStar signal is lost the playback is interrupted.

7.23.3.6 void Rovio.API.Movement.SetTuningParameters ( )

Set homing, docking and driving parameters. Refer to TuningParameters for setting the individual elements.

The parameters include: LeftRight, Forward, Reverse, DriveTurn, HomingTurn, ManDrive, ManTurn and DockTimeout.

7.23.4 Member Data Documentation

7.23.4.1 ManualDriveComponent Rovio.API.Movement.ManualDrive

Manual Drive commands.

7.23.4.2 MCUReportComponent Rovio.API.Movement.MCUReport

MCU Report commands.

7.23.4.3 ReportComponent Rovio.API.Movement.Report

Report commands.

7.23.4.4 TuningParametersComponent Rovio.API.Movement.TuningParameters

Tuning Parameters commands.

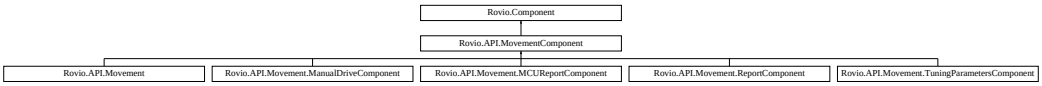
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Rovio.cs

7.24 Rovio.API.MovementComponent Class Reference

A base class for all movement commands.

Inheritance diagram for Rovio.API.MovementComponent:



Public Member Functions

- MovementComponent (Robot _robot)

  The constructor.
- virtual string Request (string value)

  Request and parse the movement command.

Protected Member Functions

- string GetParameter (string value)

  Parse the specific parameter and return its value.

Protected Attributes

- string report

  Stores the latest response from the robot. Usefull for commands supporting multiple fields in the response.

Additional Inherited Members

7.24.1 Detailed Description

A base class for all movement commands.

7.24.2 Constructor & Destructor Documentation

7.24.2.1 Rovio.API.MovementComponent.MovementComponent (Robot *_robot*)

The constructor.

7.24.3 Member Function Documentation

7.24.3.1 string Rovio.API.MovementComponent.GetParameter (string *value*) [protected]

Parse the specific parameter and return its value.

7.24.3.2 virtual string Rovio.API.MovementComponent.Request (string *value*) [virtual]

Request and parse the movement command.

Parameters

value	action value

Returns

response from the robot

7.24.4 Member Data Documentation

7.24.4.1 string Rovio.API.MovementComponent.report [protected]

Stores the latest response from the robot. Usefull for commands supporting multiple fields in the response.

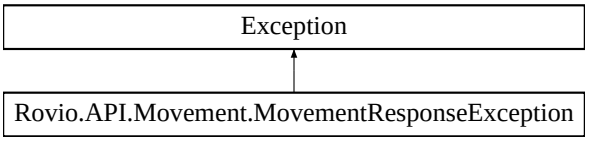
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Rovio.cs

7.25 Rovio.API.Movement.MovementResponseException Class Reference

Report errors in movement responses received from Rovio.

Inheritance diagram for Rovio.API.Movement.MovementResponseException:



Public Member Functions

- MovementResponseException (int _value)

  The constructor.

Properties

- override string Message [get]

  The error specific message.

7.25.1 Detailed Description

Report errors in movement responses received from Rovio.

7.25.2 Constructor & Destructor Documentation

7.25.2.1 Rovio.API.Movement.MovementResponseException.MovementResponseException (int *_value*)

The constructor.

7.25.3 Property Documentation

7.25.3.1 override string Rovio.API.Movement.MovementResponseException.Message [get]

The error specific message.

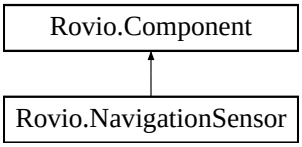
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Rovio.cs

7.26 Rovio.NavigationSensor Class Reference

A convenience class for accessing the TrueTrack navigation sensor.

Inheritance diagram for Rovio.NavigationSensor:



Public Member Functions

- NavigationSensor (Robot _robot)

  The constructor.

- override void Update ()

  Update the IRSensor value.

Additional Inherited Members

7.26.1 Detailed Description

A convenience class for accessing the TrueTrack navigation sensor.

7.26.2 Constructor & Destructor Documentation

7.26.2.1 Rovio.NavigationSensor.NavigationSensor (Robot *_robot*)

The constructor.

7.26.3 Member Function Documentation

7.26.3.1 override void Rovio.NavigationSensor.Update ( ) [virtual]

Update the IRSensor value.

Reimplemented from Rovio.Component.

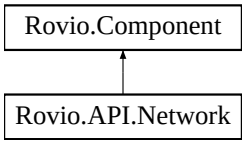
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Rovio.cs

7.27 Rovio.API.Network Class Reference

Network management.

Inheritance diagram for Rovio.API.Network:



Public Member Functions

- Network (Robot _robot)

  The constructor.
- void SetIP (string value)

  Set IP settings.
- string GetIP (string type)

  Get IP settings. Interface: eth1 wlan0
- void SetWlan (string value)

  Set Wifi settings.
- string GetWlan ()

  Get Wifi settings.
- void SetDDNS (string value)

  Set dyndns.org service Service DDNS service provider User username Pass password IP IP address (null for auto detect) Proxy name of the proxy ProxyPort port of the proxy ProxyUser username of the proxy ProxyPass password of the proxy

- string GetDDNS ()

  Get DDNS settings.
- void SetMAC (string address)

  Set MAC address.
- string GetMAC ()

  Get MAC address.

Additional Inherited Members

7.27.1 Detailed Description

Network management.

7.27.2 Constructor & Destructor Documentation

7.27.2.1 Rovio.API.Network.Network (Robot *_robot*)

The constructor.

7.27.3 Member Function Documentation

7.27.3.1 string Rovio.API.Network.GetDDNS ( )

Get DDNS settings.

Each line represents an item, and every item is in the format as Name = Value. (Refer to SetDDNS.cgi) Return information represent by Info should be one of the following values: Updated Updating Failed Updating IP Checked Not Update

7.27.3.2 string Rovio.API.Network.GetIP (string *type*)

Get IP settings. Interface: eth1 wlan0

Todo Implement the enum input parameter.

7.27.3.3 string Rovio.API.Network.GetMAC ( )

Get MAC address.

7.27.3.4 string Rovio.API.Network.GetWlan ( )

Get Wifi settings.

7.27.3.5 void Rovio.API.Network.SetDDNS (string *value*)

Set dyndns.org service Service DDNS service provider User username Pass password IP IP address (null for auto detect) Proxy name of the proxy ProxyPort port of the proxy ProxyUser username of the proxy ProxyPass password of the proxy

Set the account for dyndns.org. To connect the dyndns server, if HTTP proxy is required, set the Proxy relative value, otherwise leave them blank. If sIPAddress is not set, the device will detect the IP address automatically.

Todo Fix the input parameter according to the following format: /SetDDNS.cgi?[Enable=<true|false>][Service=<dyndns|no-ip|dnsomatic>] [&User=sUsername][&Pass=sPassword][&DomainName=sDomainName][&IP=sIPAddress] [&Proxy=sProxyServer][&ProxyPort=iProxyServerPort][&ProxyUser=sProxyUsername] [&ProxyPass=s-ProxyPassword][&RedirectUrl=sUrl]

7.27.3.6 void Rovio.API.Network.SetIP (string *value*)

Set IP settings.

Todo Fix the input value according to the following format: /SetIP.cgi?[Interface=<eth1|wlan0>][&Enable=<true|false>][&I-PWay=<manually|dhcp>][&CameraName=sName] [&IP=sIP][&Netmask=sNetmask][&Gateway=sGateway][&D-NS0=sDNS0][&DNS1=sDNS1][&DNS2=sDNS2]

7.27.3.7 void Rovio.API.Network.SetMAC (string *address*)

Set MAC address.

7.27.3.8 void Rovio.API.Network.SetWlan (string *value*)

Set Wifi settings.

Todo Fix the input parameter according to the following format: /SetWlan.cgi?[Mode=<Managed|Ad-Hoc>][&Channel=sChannel] [&ESSID=sEssid][&WepSet=<Disable|K64|K128|ASC>] [&WepAsc=s-WepAsc][&Wep64type=<Wep64HEX|Wep64ASC>][&Wep64=sWep64][&Wep128type=<Wep128HEX|-Wep128ASC>][&Wep128=sWep128]

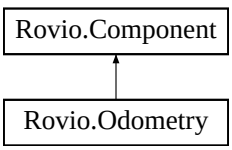
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Rovio.cs

7.28 Rovio.Odometry Class Reference

A convenience class for odometry sensor.

Inheritance diagram for Rovio.Odometry:



Public Member Functions

- Odometry (Robot _robot)

  The constructor.
- override void Update ()

  Update the odometry readings.

Properties

- int LeftWheelTicks [get]

  Left wheel ticks including rotation direction.
- int RightWheelTicks [get]

  Right wheel ticks including rotation direction.
- int RearWheelTicks [get]

  Rear wheel ticks including rotation direction.

Additional Inherited Members

7.28.1 Detailed Description

A convenience class for odometry sensor.

7.28.2 Constructor & Destructor Documentation

7.28.2.1 Rovio.Odometry.Odometry (Robot *_robot*)

The constructor.

7.28.3 Member Function Documentation

7.28.3.1 override void Rovio.Odometry.Update ( ) [virtual]

Update the odometry readings.

Reimplemented from Rovio.Component.

7.28.4 Property Documentation

7.28.4.1 int Rovio.Odometry.LeftWheelTicks [get]

Left wheel ticks including rotation direction.

7.28.4.2 int Rovio.Odometry.RearWheelTicks [get]

Rear wheel ticks including rotation direction.

7.28.4.3 int Rovio.Odometry.RightWheelTicks [get]

Right wheel ticks including rotation direction.

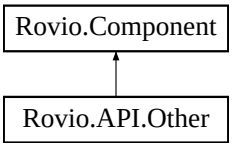
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Rovio.cs

7.29 Rovio.API.Other Class Reference

Manage camera settings, get audio and video streams, etc.

Inheritance diagram for Rovio.API.Other:



Classes

- class StatusComponent

  Return the run-time status of Rovio including camera settings, true track sensor settings, etc. Part of the Other command. Call Update() before accessing individual parameters of the report. This solution reduces the data traffic when accessing multiple parameters at the same time.

Public Member Functions

- Other (Robot _robot)

  The constructor.
- void SetCameraName (string name)

  Set camera's name.
- string GetCameraName ()

  Get camera's name.
- string GetLog ()

  Get Rovios system logs information.
- string GetVer ()

  Get Rovios base firmware version. Rovio also has a UI version and a NS2 version this function only get the base OS version.
- void SetFactoryDefault ()

  Change all settings to factory-default.
- void Reboot ()

  Reboot Rovio.
- string GetData (bool status)

  Get images/status with "multipart/x-mixed-replace" mime-type.
- string GetAudio ()

  Send audio to server and playback at server side.
- void SetMediaFormat (string audio, string video)

  Set the media format.
- string GetMediaFormat ()

  Get the media format.
- void Upload ()

  Upload new firmware image.
- void Cmd (string value)

  Use this command to combine several commands to a single http request, that is, user can call two or more commands through Cmd.cgi.

Public Attributes

- StatusComponent Status

  Status.

Additional Inherited Members

7.29.1 Detailed Description

Manage camera settings, get audio and video streams, etc.

Todo Partially implemented: GetStatus command only.

7.29.2 Constructor & Destructor Documentation

7.29.2.1 Rovio.API.Other.Other (Robot *_robot*)

The constructor.

7.29.3 Member Function Documentation

7.29.3.1 void Rovio.API.Other.Cmd (string *value*)

Use this command to combine several commands to a single http request, that is, user can call two or more commands through Cmd.cgi.

Some commands may use the same parameter's name, so the parameter's position should be in correct order. If you place parameters of sCommandName1 after sCommandName2, the behaviors of IP Camera is unexpected. If there are sub-commands that get information from Rovio, (such as GetUser.cgi, PPPoE.cgi?Action=GetInfo), "RedirectUrl" should not be specified, otherwise the browser will be redirected to the address specified by "RedirectUrl", and certainly not what you want.

7.29.3.2 string Rovio.API.Other.GetAudio ( )

Send audio to server and playback at server side.

The data flow is from client to IPCam, which is different from GetData.cgi. The audio data must be send with HTTP POST method. Audio format: 16bit PCM, 8000Hz

7.29.3.3 string Rovio.API.Other.GetCameraName ( )

Get camera's name.

7.29.3.4 string Rovio.API.Other.GetData (bool *status*)

Get images/status with "multipart/x-mixed-replace" mime-type.

GetData.cgi is designed for "server-push". "Server-push" means that user need not always detect camera's state, and the camera server transfers the camera data on its own.

7.29.3.5 string Rovio.API.Other.GetLog ( )

Get Rovios system logs information.

7.29.3.6 string Rovio.API.Other.GetMediaFormat ( )

Get the media format.

Audio 0 AMR 1 PCM 2 IMAADPCM 3 ULAW 4 ALAW

Video 1 H263 2 MPEG4

7.29.3.7 string Rovio.API.Other.GetVer ( )

Get Rovios base firmware version. Rovio also has a UI version and a NS2 version this function only get the base OS version.

7.29.3.8 void Rovio.API.Other.Reboot ( )

Reboot Rovio.

7.29.3.9 void Rovio.API.Other.SetCameraName (string *name*)

Set camera's name.

7.29.3.10 void Rovio.API.Other.SetFactoryDefault ( )

Change all settings to factory-default.

7.29.3.11 void Rovio.API.Other.SetMediaFormat (string *audio*, string *video*)

Set the media format.

Audio = 0 4 Video = 0 1

7.29.3.12 void Rovio.API.Other.Upload ( )

Upload new firmware image.

7.29.4 Member Data Documentation

7.29.4.1 StatusComponent Rovio.API.Other.Status

Status.

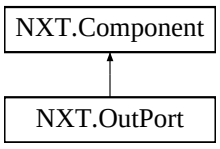
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Rovio.cs

7.30 NXT.OutPort Class Reference

Output port.

Inheritance diagram for NXT.OutPort:



Public Member Functions

- OutPort (Robot _robot)

  The constructor.
- void Update ()

  Update the output port state.
- void ResetMotorPosition (bool relative)

  Reset motor position (absolute/relative)

Public Attributes

- int Number = 0

  Port number.

Properties

- int TachoCount `[get]`

  Number of counts since the last reset of the motor counter.
- int BlockTachoCount `[get]`

  Current position relative to last programmed movement
- int RotationCount `[get]`

  Current position relative to last reset of the rotation sensor.
- int PowerSetpoint `[get, set]`

  Power setpoint.
- OutputMode Mode `[get, set]`

  Output mode.
- NXT.RegulationMode RegulationMode `[get, set]`

  Regulation mode.
- int TurnRatio `[get, set]`

  Turn ratio.
- NXT.RunState RunState `[get, set]`

  Run state.
- int TachoLimit `[get, set]`

  Tacho limit.

Additional Inherited Members

7.30.1 Detailed Description

Output port.

7.30.2 Constructor & Destructor Documentation

7.30.2.1 NXT.OutPort.OutPort (Robot *_robot*)

The constructor.

7.30.3 Member Function Documentation

7.30.3.1 void NXT.OutPort.ResetMotorPosition (bool *relative*)

Reset motor position (absolute/relative)

7.30.3.2 void NXT.OutPort.Update ( )

Update the output port state.

7.30.4 Member Data Documentation

7.30.4.1 int NXT.OutPort.Number = 0

Port number.

7.30.5 Property Documentation

7.30.5.1 int NXT.OutPort.BlockTachoCount [get]

Current position relative to last programmed movement

7.30.5.2 OutputMode NXT.OutPort.Mode [get], [set]

Output mode.

7.30.5.3 int NXT.OutPort.PowerSetpoint [get], [set]

Power setpoint.

7.30.5.4 NXT.RegulationMode NXT.OutPort.RegulationMode [get], [set]

Regulation mode.

7.30.5.5 int NXT.OutPort.RotationCount [get]

Current position relative to last reset of the rotation sensor.

7.30.5.6 NXT.RunState NXT.OutPort.RunState [get], [set]

Run state.

7.30.5.7 int NXT.OutPort.TachoCount [get]

Number of counts since the last reset of the motor counter.

7.30.5.8 int NXT.OutPort.TachoLimit [get], [set]

Tacho limit.

7.30.5.9 int NXT.OutPort.TurnRatio [get], [set]

Turn ratio.

The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/NXT.cs

7.31 Roomba.Leds.PowerLed Class Reference

Power Led - controls the colour and intensity. iCreate left led.

Public Member Functions

- PowerLed (Leds _leds)

  The constructor.

Properties

- int Color [get, set]

  Power Led Colour - 0-255 (green-red).
- int Intensity [get, set]

  Power Led Intensity - 0-255 (off-on).

7.31.1 Detailed Description

Power Led - controls the colour and intensity. iCreate left led.

7.31.2 Constructor & Destructor Documentation

7.31.2.1 Roomba.Leds.PowerLed.PowerLed (Leds *_leds*)

The constructor.

7.31.3 Property Documentation

7.31.3.1 int Roomba.Leds.PowerLed.Color [get], [set]

Power Led Colour - 0-255 (green-red).

7.31.3.2 int Roomba.Leds.PowerLed.Intensity [get], [set]

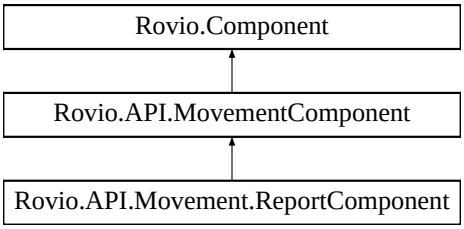
Power Led Intensity - 0-255 (off-on).

The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Roomba.cs

7.32 Rovio.API.Movement.ReportComponent Class Reference

Inheritance diagram for Rovio.API.Movement.ReportComponent:



Public Member Functions

- ReportComponent (Robot _robot)

  Contructor.
- override void Update ()

  Update the report.

Properties

- double X `[get]`

  Average X location of Rovio in relation to the strongest beacon.
- double Y `[get]`

  Average Y location of Rovio in relation to the strongest beacon.
- double Theta `[get]`

  Average orientation of Rovio in relation to the strongest beacon.
- int RoomID `[get]`

  Room ID: 0: home base, 1-9: mutable room ID.
- int NavigationSS `[get]`

  Navigation signal strength. %> 47000: strong signal %< 5000: no signal
- int BeaconSS `[get]`

  Signal strength for docking beacon. (0 - 65535)
- int BeaconX `[get]`

  Horizontal position of a beacon (as seen by NorthStar). (-32767 - 32768)
- int NextRoomID `[get]`

  The next strongest room beacon ID. -1: no room found, 1-9: mutable room ID.
- int NextRoomSS `[get]`

  The next strongest room beacon signal. (0 - 65535)

  47000: strong signal

  < 5000: no signal
- Movement.NavigationState Status `[get]`

  Rovio status.
- int UIStatus `[get]`

  Mysterious and non-documented variable: UserInterface status?
- int Resistance `[get]`

  Status of robot resistance to drive into areas badly covered by NorthStar. NOT IN USE!
- int StateMachine `[get]`

  Current status of the navigation state machine.
- int WayPoint `[get]`

Current waypoint on the path. (1 - 10)

- int Flags [get]

  Flags: 1: home position, 2: obstacle detected, 3: IR detector activated
- int Brightness [get]

  Camera brightness level.
- Camera.ImageResolution Resolution [get]

  Camera resolution.
- Camera.ImageCompression Compression [get]

  Camera compression ratio.
- int Framerate [get]

  Camera frame rate.
- int Privilege [get]

  User privilige settings. 0: admin, 1: guest
- int UserCheck [get]

  Authentication: 0: requires user/password 1: does not require user/password
- int SpeakerVolume [get]

  Speaker volume.
- int MicVolume [get]

  Micorpohne volume.
- int WifiSS [get]

  Wifi signal strength. (0 - 254)
- int ShowTime [get]

  Time display in the image: 0: off 1: on
- int DDNSState [get]

  DDNS update status: 0: no update, 1: updating, 2: update successful, 3: update failed
- int EmailState [get]

  Email update status. NOT IN USE!
- int BatteryLevel [get]

  Battery level: < 100: turn itself off, 100-106: go back home, 106-127: normal
- int Charging [get]

  Charging status: 0-79: not charging 80: charging
- int HeadPosition [get]

  Head position: 204: low 135-140: mid 65: high
- Camera.CameraFlickerFrequency FlickerFrequency [get]

  Camera flicker frequency.

Additional Inherited Members

7.32.1 Detailed Description

Report class.

Todo 'ui_status' is not documented,

Use the defined enums for popular commands (e.g. Resolution)

7.32.2 Constructor & Destructor Documentation

7.32.2.1 Rovio.API.Movement.ReportComponent.ReportComponent (Robot *_robot*)

Contructor.

7.32.3 Member Function Documentation

7.32.3.1 override void Rovio.API.Movement.ReportComponent.Update ( ) [virtual]

Update the report.

Reimplemented from Rovio.Component.

7.32.4 Property Documentation

7.32.4.1 int Rovio.API.Movement.ReportComponent.BatteryLevel [get]

Battery level: < 100: turn itself off, 100-106: go back home, 106-127: normal

7.32.4.2 int Rovio.API.Movement.ReportComponent.BeaconSS [get]

Signal strength for docking beacon. (0 - 65535)

7.32.4.3 int Rovio.API.Movement.ReportComponent.BeaconX [get]

Horizontal position of a beacon (as seen by NorthStar). (-32767 - 32768)

7.32.4.4 int Rovio.API.Movement.ReportComponent.Brightness [get]

Camera brightness level.

7.32.4.5 int Rovio.API.Movement.ReportComponent.Charging [get]

Charging status: 0-79: not charging 80: charging

7.32.4.6 Camera.ImageCompression Rovio.API.Movement.ReportComponent.Compression [get]

Camera compression ratio.

7.32.4.7 int Rovio.API.Movement.ReportComponent.DDNSState [get]

DDNS update status: 0: no update, 1: updating, 2: update successful, 3: update failed

7.32.4.8 int Rovio.API.Movement.ReportComponent.EmailState [get]

Email update status. NOT IN USE!

7.32.4.9 int Rovio.API.Movement.ReportComponent.Flags [get]

Flags: 1: home position, 2: obstacle detected, 3: IR detector activated

7.32.4.10 Camera.CameraFlickerFrequency Rovio.API.Movement.ReportComponent.FlickerFrequency [get]

Camera flicker frequency.

7.32.4.11 int Rovio.API.Movement.ReportComponent.Framerate [get]

Camera frame rate.

7.32.4.12 int Rovio.API.Movement.ReportComponent.HeadPosition [get]

Head position: 204: low 135-140: mid 65: high

7.32.4.13 int Rovio.API.Movement.ReportComponent.MicVolume [get]

Micorpohne volume.

7.32.4.14 int Rovio.API.Movement.ReportComponent.NavigationSS [get]

Navigation signal strength. %> 47000: strong signal %< 5000: no signal

7.32.4.15 int Rovio.API.Movement.ReportComponent.NextRoomID [get]

The next strongest room beacon ID. -1: no room found, 1-9: mutable room ID.

7.32.4.16 int Rovio.API.Movement.ReportComponent.NextRoomSS [get]

The next strongest room beacon signal. (0 - 65535)

47000: strong signal

< 5000: no signal

7.32.4.17 int Rovio.API.Movement.ReportComponent.Privilege [get]

User privilige settings. 0: admin, 1: guest

7.32.4.18 int Rovio.API.Movement.ReportComponent.Resistance [get]

Status of robot resistance to drive into areas badly covered by NorthStar. NOT IN USE!

7.32.4.19 Camera.ImageResolution Rovio.API.Movement.ReportComponent.Resolution [get]

Camera resolution.

7.32.4.20 int Rovio.API.Movement.ReportComponent.RoomID [get]

Room ID: 0: home base, 1-9: mutable room ID.

7.32.4.21 int Rovio.API.Movement.ReportComponent.ShowTime [get]

Time display in the image: 0: off 1: on

7.32.4.22 int Rovio.API.Movement.ReportComponent.SpeakerVolume [get]

Speaker volume.

7.32.4.23 int Rovio.API.Movement.ReportComponent.StateMachine [get]

Current status of the navigation state machine.

7.32.4.24 Movement.NavigationState Rovio.API.Movement.ReportComponent.Status [get]

Rovio status.

7.32.4.25 double Rovio.API.Movement.ReportComponent.Theta [get]

Average orientation of Rovio in relation to the strongest beacon.

7.32.4.26 int Rovio.API.Movement.ReportComponent.UIStatus [get]

Mysterious and non-documented variable: UserInterface status?

7.32.4.27 int Rovio.API.Movement.ReportComponent.UserCheck [get]

Authentication: 0: requires user/password 1: does not require user/password

7.32.4.28 int Rovio.API.Movement.ReportComponent.WayPoint [get]

Current waypoint on the path. (1 - 10)

7.32.4.29 int Rovio.API.Movement.ReportComponent.WifiSS [get]

Wifi signal strength. (0 - 254)

7.32.4.30 double Rovio.API.Movement.ReportComponent.X [get]

Average X location of Rovio in relation to the strongest beacon.

7.32.4.31 double Rovio.API.Movement.ReportComponent.Y [get]

Average Y location of Rovio in relation to the strongest beacon.

The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Rovio.cs

7.33 NXT.Robot Class Reference

The main robot class.

Classes

- class CommunicationException

  Communication exceptions with NXT.

Public Member Functions

- Robot ()

  The constructor.
- void Connect (string port_name)

  Connect to the robot.
- void Disconnect ()

  Disconnect from the robot.
- void AutoConnect ()

  Automatically scan all existing serial ports and connect to the first visible robot.
- void Send (byte[] command, out byte[] response)

  Generic send command.
- void Send (byte[] command)

  Send command without expecting the response.

Public Attributes

- API.API API

  API commands.
- DeviceInfo DeviceInfo

  Device info.
- InPort InPort1

  Input ports.
- OutPort OutPortA

  Output ports.

7.33.1 Detailed Description

The main robot class.

7.33.2 Constructor & Destructor Documentation

7.33.2.1 NXT.Robot.Robot ( )

The constructor.

7.33.3 Member Function Documentation

7.33.3.1 void NXT.Robot.AutoConnect ( )

Automatically scan all existing serial ports and connect to the first visible robot.

It is slower than directly connecting to a specific port.

7.33.3.2 void NXT.Robot.Connect (string *port_name*)

Connect to the robot.

Parameters

port_name	serial port name

7.33.3.3 void NXT.Robot.Disconnect ( )

Disconnect from the robot.

7.33.3.4 void NXT.Robot.Send (byte[] *command,* out byte[] *response*)

Generic send command.

7.33.3.5 void NXT.Robot.Send (byte[] *command*)

Send command without expecting the response.

7.33.4 Member Data Documentation

7.33.4.1 API.API NXT.Robot.API

API commands.

7.33.4.2 DeviceInfo NXT.Robot.DeviceInfo

Device info.

7.33.4.3 InPort NXT.Robot.InPort1

Input ports.

7.33.4.4 OutPort NXT.Robot.OutPortA

Output ports.

The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/NXT.cs

7.34 Roomba.Robot Class Reference

The main class for communication with Roomba through a serial interface.

Public Member Functions

- void Connect (string port_name)

  Connect to the robot.

- void Disconnect ()

  Disconnect from the robot.
- void AutoConnect ()

  Automatically scan all existing serial ports and connect to the first visible robot.
- void Send (byte[] command)

  Send a byte array corresponding to a desired command.
- void Receive (ref byte[] data, int offset, int count)

  Receive data, with timeout functionality. The call to this method has to be preceded by an appropriate request (see the Send function).

Public Attributes

- SCI.SCI SCI

  All SCI commands.
- Leds Leds

  LED related commands.
- Sensors Sensors

  Sensor realted commands.

7.34.1 Detailed Description

The main class for communication with Roomba through a serial interface.

Todo Introduce helper classes for Roomba state, odometry, motors, etc. Implement iCreate specific methods.

7.34.2 Member Function Documentation

7.34.2.1 void Roomba.Robot.AutoConnect ( )

Automatically scan all existing serial ports and connect to the first visible robot.

It is slower than directly connecting to a specific port.

7.34.2.2 void Roomba.Robot.Connect (string *port_name*)

Connect to the robot.

Parameters

port_name	serial port name

7.34.2.3 void Roomba.Robot.Disconnect ( )

Disconnect from the robot.

7.34.2.4 void Roomba.Robot.Receive (ref byte[] *data*, int *offset*, int *count*)

Receive data, with timeout functionality. The call to this method has to be preceded by an appropriate request (see the Send function).

Parameters

data	
offset	
count	

7.34.2.5 void Roomba.Robot.Send (byte[] *command*)

Send a byte array corresponding to a desired command.

Parameters

command	

7.34.3 Member Data Documentation

7.34.3.1 Leds Roomba.Robot.Leds

LED related commands.

7.34.3.2 SCI.SCI Roomba.Robot.SCI

All SCI commands.

7.34.3.3 Sensors Roomba.Robot.Sensors

Sensor realted commands.

The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Roomba.cs

7.35 Rovio.Robot Class Reference

The main class for communication with Rovio through a web client.

Public Member Functions

- Robot (string address, string user, string password)

  The constructor.
- string Request (string request)

  Request a response from the robot.
- Stream StreamRequest (string request)

  Request a stream response from the robot. Used by GetImage.cgi command.

Public Attributes

- API.API API

  API commands.
- Drive Drive

  Drive commands.

- Camera Camera

  Camera commands.
- Odometry Odometry

  Odometry commands.
- IRSensor IRSensor

  IR Sensor commands.
- NavigationSensor NavigationSensor

  Navigation sensor commands.

7.35.1 Detailed Description

The main class for communication with Rovio through a web client.

7.35.2 Constructor & Destructor Documentation

7.35.2.1 Rovio.Robot.Robot (string *address,* string *user,* string *password*)

The constructor.

7.35.3 Member Function Documentation

7.35.3.1 string Rovio.Robot.Request (string *request*)

Request a response from the robot.

Parameters

request	Request string.

Returns

Response string.

7.35.3.2 Stream Rovio.Robot.StreamRequest (string *request*)

Request a stream response from the robot. Used by GetImage.cgi command.

Parameters

request	Request string.

Returns

Response stream.

7.35.4 Member Data Documentation

7.35.4.1 API.API Rovio.Robot.API

API commands.

7.35.4.2 Camera Rovio.Robot.Camera

Camera commands.

7.35.4.3 Drive Rovio.Robot.Drive

Drive commands.

7.35.4.4 IRSensor Rovio.Robot.IRSensor

IR Sensor commands.

7.35.4.5 NavigationSensor Rovio.Robot.NavigationSensor

Navigation sensor commands.

7.35.4.6 Odometry Rovio.Robot.Odometry

Odometry commands.

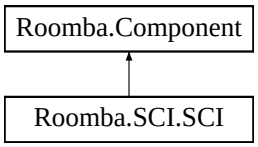
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Rovio.cs

7.36 Roomba.SCI.SCI Class Reference

Implementation of all SCI commands.

Inheritance diagram for Roomba.SCI.SCI:



Public Member Functions

- SCI (Robot _robot)

  The constructor.
- void Start ()

  Starts the SCI. The Start command must be sent before any other SCI commands. This command puts the SCI in passive mode.
- void Baud (BaudRate baud_rate)

  Sets the baud rate in bits per second (bps) at which SCI commands and data are sent according to the baud code sent in the data byte. The default baud rate at power up is 57600 bps. (See Serial Port Settings, above.) Once the baud rate is changed, it will persist until Roomba is power cycled by removing the battery (or until the battery voltage falls below the minimum required for processor operation). You must wait 100ms after sending this command before sending additional commands at the new baud rate. The SCI must be in passive, safe, or full mode to accept this command. This command puts the SCI in passive mode.
- void Control ()

Enables user control of Roomba. This command must be sent after the start command and before any control commands are sent to the SCI. The SCI must be in passive mode to accept this command. This command puts the SCI in safe mode.

- void Safe ()

  This command puts the SCI in safe mode. The SCI must be in full mode to accept this command. Note: In order to go from passive mode to safe mode, use the Control command.

- void Full ()

  Enables unrestricted control of Roomba through the SCI and turns off the safety features. The SCI must be in safe mode to accept this command. This command puts the SCI in full mode.

- void Sleep ()

  Puts Roomba to sleep, the same as a normal "power" button press. The Device Detect line must be held low for 500 ms to wake up Roomba from sleep. The SCI must be in safe or full mode to accept this command. This command puts the SCI in passive mode.

- void Spot ()

  Starts a spot cleaning cycle, the same as a normal "spot" button press. The SCI must be in safe or full mode to accept this command. This command puts the SCI in passive mode.

- void Clean ()

  Starts a normal cleaning cycle, the same as a normal "clean" button press. The SCI must be in safe or full mode to accept this command. This command puts the SCI in passive mode.

- void Max ()

  Starts a maximum time cleaning cycle, the same as a normal "max" button press. The SCI must be in safe or full mode to accept this command. This command puts the SCI in passive mode.

- void Drive (short velocity, short radius)

  Controls Roomba's drive wheels. The command takes four data bytes, which are interpreted as two 16 bit signed values using twos-complement. The first two bytes specify the average velocity of the drive wheels in millimeters per second (mm/s), with the high byte sent first. The next two bytes specify the radius, in millimeters, at which Roomba should turn. The longer radii make Roomba drive straighter; shorter radii make it turn more. A Drive command with a positive velocity and a positive radius will make Roomba drive forward while turning toward the left. A negative radius will make it turn toward the right. Special cases for the radius make Roomba turn in place or drive straight, as specified below. The SCI must be in safe or full mode to accept this command. This command does change the mode.

- void Motors (bool main_brush, bool vacuum, bool side_brush)

  Controls Roomba's cleaning motors. The state of each motor is specified by one bit in the data byte. The SCI must be in safe or full mode to accept this command. This command does not change the mode.

- void Leds (int led_bits, int power_color, int power_intensity)

  Controls Roomba's LEDs. The state of each of the spot, clean, max, and dirt detect LEDs is specified by one bit in the first data byte. The color of the status LED is specified by two bits in the first data byte. The power LED is specified by two data bytes, one for the color and one for the intensity. The SCI must be in safe or full mode to accept this command. This command does not change the mode.

- void Song (int song_number, byte[] song)

  Specifies a song to the SCI to be played later. Each song is associated with a song number which the Play command uses to select the song to play. Users can specify up to 16 songs with up to 16 notes per song. Each note is specified by a note number using MIDI note definitions and a duration specified in fractions of a second. The number of data bytes varies depending on the length of the song specified. A one note song is specified by four data bytes. For each additional note, two data bytes must be added. The SCI must be in passive, safe, or full mode to accept this command. This command does not change the mode.

- void Play (byte song_number)

  Plays one of 16 songs, as specified by an earlier Song command. If the requested song has not been specified yet, the Play command does nothing. The SCI must be in safe or full mode to accept this command. This command does not change the mode.

- void Sensors (SensorPacket packet)

  Requests the SCI to send a packet of sensor data bytes. The user can select one of four different sensor packets. The sensor data packets are explained in more detail in the next section. The SCI must be in passive, safe, or full mode to accept this command. This command does not change the mode.

- void ForceSeekingDock ()

Turns on force-seeking-dock mode, which causes the robot to immediately attempt to dock during its cleaning cycle if it encounters the docking beams from the Home Base. (Note, however, that if the robot was not active in a clean, spot or max cycle it will not attempt to execute the docking.) Normally the robot attempts to dock only if the cleaning cycle has completed or the battery is nearing depletion. This command can be sent anytime, but the mode will be cancelled if the robot turns off, begins charging, or is commanded into SCI safe or full modes.

- void UpdateSensorState (SensorPacket packet)

  Update the sensor state. The robot will send back one of four different sensor data packets in response to a Sensor command, depending on the value of the packet code data byte. The data bytes are specified below in the order in which they will be sent. A packet code value of 0 sends all of the data bytes. A value of 1 through 3 sends a subset of the sensor data. Some of the sensor data values are 16 bit values. These values are sent as two bytes, high byte first.

Public Attributes

- byte[] sensor_state

  The raw sensor data.

Additional Inherited Members

7.36.1 Detailed Description

Implementation of all SCI commands.

7.36.2 Constructor & Destructor Documentation

7.36.2.1 Roomba.SCI.SCI (Robot *_robot*)

The constructor.

7.36.3 Member Function Documentation

7.36.3.1 void Roomba.SCI.SCI.Baud (BaudRate *baud_rate*)

Sets the baud rate in bits per second (bps) at which SCI commands and data are sent according to the baud code sent in the data byte. The default baud rate at power up is 57600 bps. (See Serial Port Settings, above.) Once the baud rate is changed, it will persist until Roomba is power cycled by removing the battery (or until the battery voltage falls below the minimum required for processor operation). You must wait 100ms after sending this command before sending additional commands at the new baud rate. The SCI must be in passive, safe, or full mode to accept this command. This command puts the SCI in passive mode.

Parameters

baud_rate	baud rate

7.36.3.2 void Roomba.SCI.SCI.Clean ( )

Starts a normal cleaning cycle, the same as a normal “clean” button press. The SCI must be in safe or full mode to accept this command. This command puts the SCI in passive mode.

7.36.3.3 void Roomba.SCI.SCI.Control ( )

Enables user control of Roomba. This command must be sent after the start command and before any control commands are sent to the SCI. The SCI must be in passive mode to accept this command. This command puts the

SCI in safe mode.

7.36.3.4 void Roomba.SCI.SCI.Drive (short *velocity,* short *radius*)

Controls Roomba's drive wheels. The command takes four data bytes, which are interpreted as two 16 bit signed values using twos-complement. The first two bytes specify the average velocity of the drive wheels in millimeters per second (mm/s), with the high byte sent first. The next two bytes specify the radius, in millimeters, at which Roomba should turn. The longer radii make Roomba drive straighter; shorter radii make it turn more. A Drive command with a positive velocity and a positive radius will make Roomba drive forward while turning toward the left. A negative radius will make it turn toward the right. Special cases for the radius make Roomba turn in place or drive straight, as specified below. The SCI must be in safe or full mode to accept this command. This command does change the mode.

7.36.3.5 void Roomba.SCI.SCI.ForceSeekingDock ( )

Turns on force-seeking-dock mode, which causes the robot to immediately attempt to dock during its cleaning cycle if it encounters the docking beams from the Home Base. (Note, however, that if the robot was not active in a clean, spot or max cycle it will not attempt to execute the docking.) Normally the robot attempts to dock only if the cleaning cycle has completed or the battery is nearing depletion. This command can be sent anytime, but the mode will be cancelled if the robot turns off, begins charging, or is commanded into SCI safe or full modes.

7.36.3.6 void Roomba.SCI.SCI.Full ( )

Enables unrestricted control of Roomba through the SCI and turns off the safety features. The SCI must be in safe mode to accept this command. This command puts the SCI in full mode.

7.36.3.7 void Roomba.SCI.SCI.Leds (int *led_bits,* int *power_color,* int *power_intensity*)

Controls Roomba's LEDs. The state of each of the spot, clean, max, and dirt detect LEDs is specified by one bit in the first data byte. The color of the status LED is specified by two bits in the first data byte. The power LED is specified by two data bytes, one for the color and one for the intensity. The SCI must be in safe or full mode to accept this command. This command does not change the mode.

Parameters

led_bits	
power_color	
power_intensity	

7.36.3.8 void Roomba.SCI.SCI.Max ( )

Starts a maximum time cleaning cycle, the same as a normal "max" button press. The SCI must be in safe or full mode to accept this command. This command puts the SCI in passive mode.

7.36.3.9 void Roomba.SCI.SCI.Motors (bool *main_brush,* bool *vacuum,* bool *side_brush*)

Controls Roomba's cleaning motors. The state of each motor is specified by one bit in the data byte. The SCI must be in safe or full mode to accept this command. This command does not change the mode.

Parameters

main_brush	Main brush on/off (true/false)
vacuum	Vacuum on/off (true/false)
side_brush	Side brush on/off (true/false)

7.36.3.10 void Roomba.SCI.SCI.Play (byte *song_number*)

Plays one of 16 songs, as specified by an earlier Song command. If the requested song has not been specified yet, the Play command does nothing. The SCI must be in safe or full mode to accept this command. This command does not change the mode.

7.36.3.11 void Roomba.SCI.SCI.Safe ( )

This command puts the SCI in safe mode. The SCI must be in full mode to accept this command. Note: In order to go from passive mode to safe mode, use the Control command.

7.36.3.12 void Roomba.SCI.SCI.Sensors (SensorPacket *packet*)

Requests the SCI to send a packet of sensor data bytes. The user can select one of four different sensor packets. The sensor data packets are explained in more detail in the next section. The SCI must be in passive, safe, or full mode to accept this command. This command does not change the mode.

Parameters

packet	

7.36.3.13 void Roomba.SCI.SCI.Sleep ( )

Puts Roomba to sleep, the same as a normal “power” button press. The Device Detect line must be held low for 500 ms to wake up Roomba from sleep. The SCI must be in safe or full mode to accept this command. This command puts the SCI in passive mode.

7.36.3.14 void Roomba.SCI.SCI.Song (int *song_number,* byte[] *song*)

Specifies a song to the SCI to be played later. Each song is associated with a song number which the Play command uses to select the song to play. Users can specify up to 16 songs with up to 16 notes per song. Each note is specified by a note number using MIDI note definitions and a duration specified in fractions of a second. The number of data bytes varies depending on the length of the song specified. A one note song is specified by four data bytes. For each additional note, two data bytes must be added. The SCI must be in passive, safe, or full mode to accept this command. This command does not change the mode.

Parameters

song_number	
song	

7.36.3.15 void Roomba.SCI.SCI.Spot ( )

Starts a spot cleaning cycle, the same as a normal “spot” button press. The SCI must be in safe or full mode to accept this command. This command puts the SCI in passive mode.

7.36.3.16 void Roomba.SCI.SCI.Start ( )

Starts the SCI. The Start command must be sent before any other SCI commands. This command puts the SCI in passive mode.

7.36.3.17 void Roomba.SCI.SCI.UpdateSensorState (SensorPacket *packet*)

Update the sensor state. The robot will send back one of four different sensor data packets in response to a Sensor command, depending on the value of the packet code data byte. The data bytes are specified below in the order in which they will be sent. A packet code value of 0 sends all of the data bytes. A value of 1 through 3 sends a subset of the sensor data. Some of the sensor data values are 16 bit values. These values are sent as two bytes, high byte first.

Parameters

packet	

7.36.4 Member Data Documentation

7.36.4.1 byte [ ] Roomba.SCI.SCI.sensor_state

The raw sensor data.

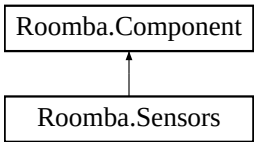
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Roomba.cs

7.37 Roomba.Sensors Class Reference

A convenience class for accessing Roomba's sensors.

Inheritance diagram for Roomba.Sensors:



Classes

- class Bump

  The state of the bump sensors: false = no bump, true = bump.
- class Button

  The state of the four Roomba buttons: false = button not pressed, true = button pressed.
- class Cliff

  The state of the cliff sensors: false = no cliff, true = cliff.
- class DirtDetector

  The current dirt detection level (0-255) of the dirt detector. Higher values indicate higher levels of dirt detected.
- class MotorOvercurrent

  The state of the five motors' overcurrent sensors: false = no overcurrent, true = overcurrent.
- class Wheeldrop

  The state of the Wheeldrop sensors: false = wheel up, true = wheel dropped.

Public Member Functions

- Sensors (Robot _robot)

  The constructor.

- void Update (SensorPacket packet)

  Update the sensor state.

Public Attributes

- Bump Bumps

  Bump sensors.
- Wheeldrop Wheeldrops

  Wheeldrop sensors.
- Cliff Cliffs

  Cliff sensors.
- MotorOvercurrent MotorOvercurrents

  Motor Overcurrent sensors.
- DirtDetector DirtDetectors

  Dirt Detector sensors.
- Button Buttons

  Buttons.

Properties

- bool Wall `[get]`

  The state of the wall sensor: false = no wall, true = wall.
- bool VirtualWall `[get]`

  The state of the virtual wall sensor: false = no wall, true = wall.
- int RemoteControlCommand `[get]`

  The command number of the remote control command currently being received by Roomba. A value of 255 indicates that no remote control command is being received.
- int Distance `[get]`

  The distance that Roomba has traveled in millimeters since the distance it was last requested. This is the same as the sum of the distance traveled by both wheels divided by two. Positive values indicate travel in the forward direction; negative in the reverse direction. If the value is not polled frequently enough, it will be capped at its minimum or maximum. (16 bit signed integer)
- int Angle `[get]`

  The angle that Roomba has turned through since the angle was last requested. The angle is expressed as the difference in the distance traveled by Roomba's two wheels in millimeters, specifically the right wheel distance minus the left wheel distance, divided by two. This makes counter-clockwise angles positive and clockwise angles negative. This can be used to directly calculate the angle that Roomba has turned through since the last request. Since the distance between Roomba's wheels is 258mm, the equations for calculating the angles in familiar units are: Angle in radians = (2 ∗ difference) / 258 Angle in degrees = (360 ∗ difference) / (258 ∗ Pi). If the value is not polled frequently enough, it will be capped at its minimum or maximum. (16 bit signed integer)
- ChargingState ChargingState `[get]`

  Get charging state of the robot.
- int Voltage `[get]`

  The voltage of Roomba's battery in millivolts (mV).
- int Current `[get]`

  The current in milliamps (mA) flowing into or out of Roomba's battery. Negative currents indicate current is flowing out of the battery, as during normal running. Positive currents indicate current is flowing into the battery, as during charging.
- int Temperature `[get]`

  The temperature of Roomba's battery in degrees Celsius.
- int Charge `[get]`

  The current charge of Roomba's battery in milliamp-hours (mAh). The charge value decreases as the battery is depleted during running and increases when the battery is charged.

- int Capacity [get]

  The estimated charge capacity of Roomba's battery. When the Charge value reaches the Capacity value, the battery is fully charged.

Additional Inherited Members

7.37.1 Detailed Description

A convenience class for accessing Roomba's sensors.

7.37.2 Constructor & Destructor Documentation

7.37.2.1 Roomba.Sensors.Sensors (Robot *_robot*)

The constructor.

7.37.3 Member Function Documentation

7.37.3.1 void Roomba.Sensors.Update (SensorPacket *packet*)

Update the sensor state.

Parameters

packet	

7.37.4 Member Data Documentation

7.37.4.1 Bump Roomba.Sensors.Bumps

Bump sensors.

7.37.4.2 Button Roomba.Sensors.Buttons

Buttons.

7.37.4.3 Cliff Roomba.Sensors.Cliffs

Cliff sensors.

7.37.4.4 DirtDetector Roomba.Sensors.DirtDetectors

Dirt Detector sensors.

7.37.4.5 MotorOvercurrent Roomba.Sensors.MotorOvercurrents

Motor Overcurrent sensors.

7.37.4.6 Wheeldrop Roomba.Sensors.Wheeldrops

Wheeldrop sensors.

7.37.5 Property Documentation

7.37.5.1 int Roomba.Sensors.Angle [get]

The angle that Roomba has turned through since the angle was last requested. The angle is expressed as the difference in the distance traveled by Roomba's two wheels in millimeters, specifically the right wheel distance minus the left wheel distance, divided by two. This makes counter-clockwise angles positive and clockwise angles negative. This can be used to directly calculate the angle that Roomba has turned through since the last request. Since the distance between Roomba's wheels is 258mm, the equations for calculating the angles in familiar units are: Angle in radians = (2 $*$ difference) / 258 Angle in degrees = (360 $*$ difference) / (258 $*$ Pi). If the value is not polled frequently enough, it will be capped at its minimum or maximum. (16 bit signed integer)

7.37.5.2 int Roomba.Sensors.Capacity [get]

The estimated charge capacity of Roomba's battery. When the Charge value reaches the Capacity value, the battery is fully charged.

7.37.5.3 int Roomba.Sensors.Charge [get]

The current charge of Roomba's battery in milliamp-hours (mAh). The charge value decreases as the battery is depleted during running and increases when the battery is charged.

7.37.5.4 ChargingState Roomba.Sensors.ChargingState [get]

Get charging state of the robot.

7.37.5.5 int Roomba.Sensors.Current [get]

The current in milliamps (mA) flowing into or out of Roomba's battery. Negative currents indicate current is flowing out of the battery, as during normal running. Positive currents indicate current is flowing into the battery, as during charging.

7.37.5.6 int Roomba.Sensors.Distance [get]

The distance that Roomba has traveled in millimeters since the distance it was last requested. This is the same as the sum of the distance traveled by both wheels divided by two. Positive values indicate travel in the forward direction; negative in the reverse direction. If the value is not polled frequently enough, it will be capped at its minimum or maximum. (16 bit signed integer)

7.37.5.7 int Roomba.Sensors.RemoteControlCommand [get]

The command number of the remote control command currently being received by Roomba. A value of 255 indicates that no remote control command is being received.

7.37.5.8 int Roomba.Sensors.Temperature [get]

The temperature of Roomba's battery in degrees Celsius.

7.37.5.9 bool Roomba.Sensors.VirtualWall [get]

The state of the virtual wall sensor: false = no wall, true = wall.

7.37.5.10 int Roomba.Sensors.Voltage [get]

The voltage of Roomba's battery in millivolts (mV).

7.37.5.11 bool Roomba.Sensors.Wall [get]

The state of the wall sensor: false = no wall, true = wall.

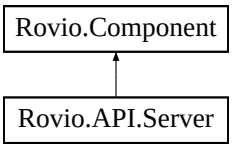
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Roomba.cs

7.38 Rovio.API.Server Class Reference

Manage server settings.

Inheritance diagram for Rovio.API.Server:



Public Member Functions

- Server (Robot _robot)
  The constructor.
- void SetHTTP (string value)
  Set the parameters for HTTP server (Currently only TCP port).
- void SetHTTP (string port0, string port1)
  Set the parameters for HTTP server (Currently only TCP port).
- string GetHTTP ()
  Get HTTP server's settings.

Additional Inherited Members

7.38.1 Detailed Description

Manage server settings.

7.38.2 Constructor & Destructor Documentation

7.38.2.1 Rovio.API.Server.Server (Robot *_robot*)

The constructor.

7.38.3 Member Function Documentation

7.38.3.1 string Rovio.API.Server.GetHTTP ( )

Get HTTP server's settings.

7.38.3.2 void Rovio.API.Server.SetHTTP (string *value*)

Set the parameters for HTTP server (Currently only TCP port).

Modyfing one port.

7.38.3.3 void Rovio.API.Server.SetHTTP (string *port0,* string *port1*)

Set the parameters for HTTP server (Currently only TCP port).

Modyfing two ports.

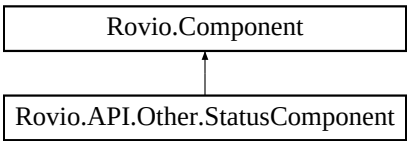
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Rovio.cs

7.39 Rovio.API.Other.StatusComponent Class Reference

Return the run-time status of Rovio including camera settings, true track sensor settings, etc. Part of the Other command. Call Update() before accessing individual parameters of the report. This solution reduces the data traffic when accessing multiple parameters at the same time.

Inheritance diagram for Rovio.API.Other.StatusComponent:



Public Member Functions

- StatusComponent (Robot _robot)
  The constructor.
- override void Update ()
  Update the report.

Properties

- int CameraState `[get]`
  Camera state: 0 = off, 1 = on.
- int ModemState `[get]`
  Modem state: 0 = off, 1 = on line(common mode), 2 = connecting(common mode).
- int PPPoEState `[get]`
  PPPoE State - same as Modem state.
- int XDirection `[get]`
  Reserved.
- int YDirection `[get]`
  Reserved.
- int Focus `[get]`
  Reserved.
- int Brightness `[get]`
  Camera brightness: 0-255.

- int Contrast [get]

  Camera contrast: 0-255.
- Camera.ImageResolution Resolution [get]

  Camera resolution.
- Camera.ImageCompression Compression [get]

  Reserved.
- int Privilege [get]

  Priviliege: 0 = super user(administrator), 1 = common user.
- int PictureIndex [get]

  Picture index: 999999 = invalid picture.
- int EmailState [get]

  Email state: 0 = do not send motion-detected pictures, 1 = send motion-detected pictures, success, 2 = send motion-detected pictures, fail (wrong IP, user or password?).
- int UserCheck [get]

  User check: 0 = do not check user, any user can connect and act as a super user, 1 = username and password required, only username is "administrator" has the super privilege.
- int ImageFileLength [get]

  Image file length: length in bytes.
- int MonitorRect [get]

  4 bytes left(0-9999), 4 bytes - top(0-9999), 4 bytes - right(0-9999), 4 bytes - bottom(0-9999)
- int FTPState [get]

  FTP state: 0 = disable ftp upload, 1 = enable ftp upload, and upload success, 2 = enable ftp upload, but fail(wrong IP, user or password?).
- int Saturation [get]

  Camera image saturation: 0-255.
- int MotionDetectedIndex [get]

  Motion detection index: 999999 = init value.
- int Hue [get]

  Hue: 0-255.
- int Sharpness [get]

  Sharpness: 0-255.
- int MotionDetectWay [get]

  0 = no motion detection, non-zero = motion detection
- Camera.CameraFlickerFrequency FlickerFrequency [get]

  Camera's frequency: 0 = outdoor, 1 = 50Hz, 2 = 60Hz
- int ChannelMode [get]

  0 = fixed mode, 1 = round robin mode.
- int ChannelValue [get]

  In fixed mode, the value may be from 0 to 3 In round robin mode, the value may be from 1 to 15.
- int AudioVolume [get]

  Audio volume.
- int DynamicDNSState [get]

  DNS state: 0 = no update, 1 = updating, 2 = update successful, 3 = update failed.
- int AudioState [get]

  0 = audio disabled, 1 = audio enabled.
- int Framerate [get]

  Framerate
- int SpeakerVolume [get]

  Speaker volume.
- int MicVolume [get]

  Microphone volume.

- int ShowTime [get]

  0 = do not show time in image, 1 = show time in image.
- int WifiStrength [get]

  0-15, 0 is max.
- int BatteryLevel [get]

  0-255, 255 is Max.

Additional Inherited Members

7.39.1 Detailed Description

Return the run-time status of Rovio including camera settings, true track sensor settings, etc. Part of the Other command. Call Update() before accessing individual parameters of the report. This solution reduces the data traffic when accessing multiple parameters at the same time.

7.39.2 Constructor & Destructor Documentation

7.39.2.1 Rovio.API.Other.StatusComponent.StatusComponent (Robot *_robot*)

The constructor.

7.39.3 Member Function Documentation

7.39.3.1 override void Rovio.API.Other.StatusComponent.Update ( ) [virtual]

Update the report.

Reimplemented from Rovio.Component.

7.39.4 Property Documentation

7.39.4.1 int Rovio.API.Other.StatusComponent.AudioState [get]

0 = audio disabled, 1 = audio enabled.

7.39.4.2 int Rovio.API.Other.StatusComponent.AudioVolume [get]

Audio volume.

7.39.4.3 int Rovio.API.Other.StatusComponent.BatteryLevel [get]

0-255, 255 is Max.

7.39.4.4 int Rovio.API.Other.StatusComponent.Brightness [get]

Camera brightness: 0-255.

7.39.4.5 int Rovio.API.Other.StatusComponent.CameraState [get]

Camera state: 0 = off, 1 = on.

7.39.4.6 int Rovio.API.Other.StatusComponent.ChannelMode [get]

0 = fixed mode, 1 = round robin mode.

7.39.4.7 int Rovio.API.Other.StatusComponent.ChannelValue [get]

In fixed mode, the value may be from 0 to 3 In round robin mode, the value may be from 1 to 15.

7.39.4.8 Camera.ImageCompression Rovio.API.Other.StatusComponent.Compression [get]

Reserved.

7.39.4.9 int Rovio.API.Other.StatusComponent.Contrast [get]

Camera contrast: 0-255.

7.39.4.10 int Rovio.API.Other.StatusComponent.DynamicDNSState [get]

DNS state: 0 = no update, 1 = updating, 2 = update successful, 3 = update failed.

7.39.4.11 int Rovio.API.Other.StatusComponent.EmailState [get]

Email state: 0 = do not send motion-detected pictures, 1 = send motion-detected pictures, success, 2 = send motion-detected pictures, fail (wrong IP, user or password?).

7.39.4.12 Camera.CameraFlickerFrequency Rovio.API.Other.StatusComponent.FlickerFrequency [get]

Camera's frequency: 0 = outdoor, 1 = 50Hz, 2 = 60Hz

7.39.4.13 int Rovio.API.Other.StatusComponent.Focus [get]

Reserved.

7.39.4.14 int Rovio.API.Other.StatusComponent.Framerate [get]

Framerate

7.39.4.15 int Rovio.API.Other.StatusComponent.FTPState [get]

FTP state: 0 = disable ftp upload, 1 = enable ftp upload, and upload success, 2 = enable ftp upload, but fail(wrong IP, user or password?).

7.39.4.16 int Rovio.API.Other.StatusComponent.Hue [get]

Hue: 0-255.

7.39.4.17 int Rovio.API.Other.StatusComponent.ImageFileLength [get]

Image file length: length in bytes.

7.39.4.18 int Rovio.API.Other.StatusComponent.MicVolume [get]

Microphone volume.

7.39.4.19 int Rovio.API.Other.StatusComponent.ModemState [get]

Modem state: 0 = off, 1 = on line(common mode), 2 = connecting(common mode).

7.39.4.20 int Rovio.API.Other.StatusComponent.MonitorRect [get]

4 bytes left(0-9999), 4 bytes - top(0-9999), 4 bytes - right(0-9999), 4 bytes - bottom(0-9999)

Todo fix the return value

7.39.4.21 int Rovio.API.Other.StatusComponent.MotionDetectedIndex [get]

Motion detection index: 999999 = init value.

7.39.4.22 int Rovio.API.Other.StatusComponent.MotionDetectWay [get]

0 = no motion detection, non-zero = motion detection

7.39.4.23 int Rovio.API.Other.StatusComponent.PictureIndex [get]

Picture index: 999999 = invalid picture.

7.39.4.24 int Rovio.API.Other.StatusComponent.PPPoEState [get]

PPPoE State - same as Modem state.

7.39.4.25 int Rovio.API.Other.StatusComponent.Privilege [get]

Priviliege: 0 = super user(administrator), 1 = common user.

7.39.4.26 Camera.ImageResolution Rovio.API.Other.StatusComponent.Resolution [get]

Camera resolution.

7.39.4.27 int Rovio.API.Other.StatusComponent.Saturation [get]

Camera image saturation: 0-255.

7.39.4.28 int Rovio.API.Other.StatusComponent.Sharpness [get]

Sharpness: 0-255.

7.39.4.29 int Rovio.API.Other.StatusComponent.ShowTime [get]

0 = do not show time in image, 1 = show time in image.

7.39.4.30 int Rovio.API.Other.StatusComponent.SpeakerVolume [get]

Speaker volume.

7.39.4.31 int Rovio.API.Other.StatusComponent.UserCheck [get]

User check: 0 = do not check user, any user can connect and act as a super user, 1 = username and password required, only username is "administrator" has the super privilege.

7.39.4.32 int Rovio.API.Other.StatusComponent.WifiStrength [get]

0-15, 0 is max.

7.39.4.33 int Rovio.API.Other.StatusComponent.XDirection [get]

Reserved.

7.39.4.34 int Rovio.API.Other.StatusComponent.YDirection [get]

Reserved.

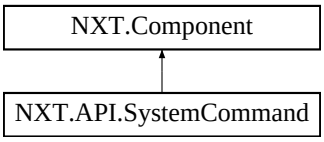
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Rovio.cs

7.40 NXT.API.SystemCommand Class Reference

System commands.

Inheritance diagram for NXT.API.SystemCommand:



Public Member Functions

- SystemCommand (Robot _robot)
  The constructor.
- void OpenRead (string name, ref int file_handle, ref int file_size)
  Open file for reading.
- void OpenWrite (string name, ref int file_handle, int file_size)
  Open file for writing.
- void Read (int file_handle, int bytes_to_read, out byte[] data)
  Read data from file.
- void Write (int file_handle, byte[] data)
  Write data to the specified file.
- void Close (int file_handle)
  Close the specified file.

- void Delete (string name)

  Delete file.
- void FindFirst (string name, ref string file_name, ref int file_handle, ref int file_size)

  Search for file
- void FindNext (int file_handle, out byte[] response)

  Search for the next file with file_handle obtained from FindFirst or Open commands.
- void GetFirmwareVersion (out byte[] response)

  Get minor and major versions of firmware and protocol.
- void OpenWriteLinear (string name, ref int file_handle, int file_size)

  Open file for writing (linear mode).
- void OpenReadLinear (string name, ref int memory_pointer)

  Open file for writing (linear mode).
- void OpenWriteData (string name, ref int file_handle, int file_size)

  Open file for writing data.
- void OpenAppendData (string name, ref int file_handle, ref int file_size)

  Open file for appending data.
- void Boot ()

  Boot the brick. USB command only.
- void SetBrickName (string name)

  Set brick name.
- void GetDeviceInfo (out byte[] response)

  Get device information in a raw format.
- void DeleteUserFlash ()

  Delete user flash memory.
- void PollLength (int buffer_type, ref int buffer_size)

  Number of bytes for the command ready in the buffer.
- void Poll (int buffer_type, int command_length, out byte[] response)

  Poll the command.
- void BluetoothFactoryReset ()

  Reset Bluetooth. USB command only.

Public Attributes

- bool RequestResponse = false

  CheckResponse will require a confirmation from NXT. It might be slower and return exceptions.

Additional Inherited Members

7.40.1 Detailed Description

System commands.

7.40.2 Constructor & Destructor Documentation

7.40.2.1 NXT.API.SystemCommand.SystemCommand (Robot *_robot*)

The constructor.

7.40.3 Member Function Documentation

7.40.3.1 void NXT.API.SystemCommand.BluetoothFactoryReset ( )

Reset Bluetooth. USB command only.

7.40.3.2 void NXT.API.SystemCommand.Boot ( )

Boot the brick. USB command only.

7.40.3.3 void NXT.API.SystemCommand.Close (int *file_handle*)

Close the specified file.

7.40.3.4 void NXT.API.SystemCommand.Delete (string *name*)

Delete file.

7.40.3.5 void NXT.API.SystemCommand.DeleteUserFlash ( )

Delete user flash memory.

7.40.3.6 void NXT.API.SystemCommand.FindFirst (string *name,* ref string *file_name,* ref int *file_handle,* ref int *file_size*)

Search for file

7.40.3.7 void NXT.API.SystemCommand.FindNext (int *file_handle,* out byte[ ] *response*)

Search for the next file with file_handle obtained from FindFirst or Open commands.

Parameters

file_handle	
response	

7.40.3.8 void NXT.API.SystemCommand.GetDeviceInfo (out byte[ ] *response*)

Get device information in a raw format.

7.40.3.9 void NXT.API.SystemCommand.GetFirmwareVersion (out byte[ ] *response*)

Get minor and major versions of firmware and protocol.

7.40.3.10 void NXT.API.SystemCommand.OpenAppendData (string *name,* ref int *file_handle,* ref int *file_size*)

Open file for appending data.

7.40.3.11 void NXT.API.SystemCommand.OpenRead (string *name,* ref int *file_handle,* ref int *file_size*)

Open file for reading.

Call Close when the file is not needed anymore.

7.40.3.12 void NXT.API.SystemCommand.OpenReadLinear (string *name,* ref int *memory_pointer*)

Open file for writing (linear mode).

7.40.3.13 void NXT.API.SystemCommand.OpenWrite (string *name,* ref int *file_handle,* int *file_size*)

Open file for writing.
Call Close when the file is not needed anymore.

7.40.3.14 void NXT.API.SystemCommand.OpenWriteData (string *name,* ref int *file_handle,* int *file_size*)

Open file for writing data.

7.40.3.15 void NXT.API.SystemCommand.OpenWriteLinear (string *name,* ref int *file_handle,* int *file_size*)

Open file for writing (linear mode).

7.40.3.16 void NXT.API.SystemCommand.Poll (int *buffer_type,* int *command_length,* out byte[] *response*)

Poll the command.

7.40.3.17 void NXT.API.SystemCommand.PollLength (int *buffer_type,* ref int *buffer_size*)

Number of bytes for the command ready in the buffer.

7.40.3.18 void NXT.API.SystemCommand.Read (int *file_handle,* int *bytes_to_read,* out byte[] *data*)

Read data from file.

7.40.3.19 void NXT.API.SystemCommand.SetBrickName (string *name*)

Set brick name.

7.40.3.20 void NXT.API.SystemCommand.Write (int *file_handle,* byte[] *data*)

Write data to the specified file.

7.40.4 Member Data Documentation

7.40.4.1 bool NXT.API.SystemCommand.RequestResponse = false

CheckResponse will require a confirmation from NXT. It might be slower and return exceptions.

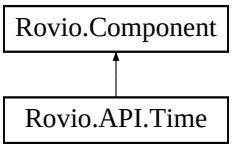
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/NXT.cs

7.41 Rovio.API.Time Class Reference

Manage time settings and time zones.

Inheritance diagram for Rovio.API.Time:



Public Member Functions

- Time (Robot _robot)

  The constructor.
- void SetTime (int seconds, int time_zone)

  Set server time zone and time. Sec1970 - seconds since "00:00:00 1/1/1970". TimeZone Time zone in minutes. (e.g. Beijing is GMT+08:00, TimeZone = -480)
- string GetTime ()

  Get current time zone and time.
- void SetLogo (string type, string position)

  Set a logo string on the image. showstring: time - time date - date ver - version pos: 0 top left 1 top right 2 bottom left 3 bottom right
- string GetLogo ()

  Get a logo string on the image.

Additional Inherited Members

7.41.1 Detailed Description

Manage time settings and time zones.

7.41.2 Constructor & Destructor Documentation

7.41.2.1 Rovio.API.Time.Time (Robot *_robot*)

The constructor.

7.41.3 Member Function Documentation

7.41.3.1 string Rovio.API.Time.GetLogo ( )

Get a logo string on the image.

/todo Fix return value. The current Rovio implementation repeats the return value twice.

7.41.3.2 string Rovio.API.Time.GetTime ( )

Get current time zone and time.

Todo Implement with DateTime input parameter.

7.41.3.3 void Rovio.API.Time.SetLogo (string *type*, string *position*)

Set a logo string on the image. showstring: time - time date - date ver - version pos: 0 top left 1 top right 2 bottom left 3 bottom right

Todo Implement the enum input parameters.

7.41.3.4 void Rovio.API.Time.SetTime (int *seconds*, int *time_zone*)

Set server time zone and time. Sec1970 - seconds since "00:00:00 1/1/1970". TimeZone Time zone in minutes. (e.g. Beijing is GMT+08:00, TimeZone = -480)

Todo Implement with DateTime input parameter.

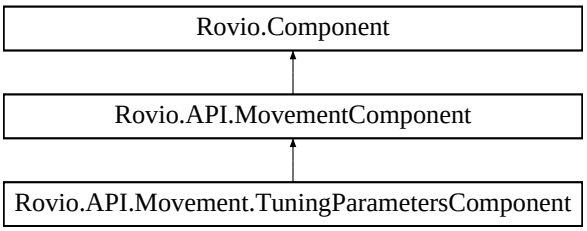
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Rovio.cs

7.42 Rovio.API.Movement.TuningParametersComponent Class Reference

Manage robot parameters used during navigation: homing, docking and automatic driving.

Inheritance diagram for Rovio.API.Movement.TuningParametersComponent:



Public Member Functions

- TuningParametersComponent (Robot _robot)
  The constructor.
- override void Update ()
  Update the report.

Properties

- int LeftRight `[get]`
  LeftRight value.
- int Forward `[get]`
  Forward value.
- int Reverse `[get]`
  Reverse value.
- int DriveTurn `[get]`
  DriveTurn value.
- int HomingTurn `[get]`

HomingTurn value.

- int ManDrive [get]

  Manual drive value.

- int ManTurn [get]

  Manual turn value.

- int DockTimeout [get]

  Dock time out.

Additional Inherited Members

7.42.1 Detailed Description

Manage robot parameters used during navigation: homing, docking and automatic driving.

7.42.2 Constructor & Destructor Documentation

7.42.2.1 Rovio.API.Movement.TuningParametersComponent.TuningParametersComponent (Robot *_robot*)

The constructor.

Parameters

_robot	

7.42.3 Member Function Documentation

7.42.3.1 override void Rovio.API.Movement.TuningParametersComponent.Update ( ) [virtual]

Update the report.

Reimplemented from Rovio.Component.

7.42.4 Property Documentation

7.42.4.1 int Rovio.API.Movement.TuningParametersComponent.DockTimeout [get]

Dock time out.

7.42.4.2 int Rovio.API.Movement.TuningParametersComponent.DriveTurn [get]

DriveTurn value.

7.42.4.3 int Rovio.API.Movement.TuningParametersComponent.Forward [get]

Forward value.

7.42.4.4 int Rovio.API.Movement.TuningParametersComponent.HomingTurn [get]

HomingTurn value.

7.42.4.5 int Rovio.API.Movement.TuningParametersComponent.LeftRight [get]

LeftRight value.

7.42.4.6 int Rovio.API.Movement.TuningParametersComponent.ManDrive [get]

Manual drive value.

7.42.4.7 int Rovio.API.Movement.TuningParametersComponent.ManTurn [get]

Manual turn value.

7.42.4.8 int Rovio.API.Movement.TuningParametersComponent.Reverse [get]

Reverse value.

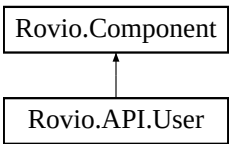
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Rovio.cs

7.43 Rovio.API.User Class Reference

Manage user accounts.

Inheritance diagram for Rovio.API.User:



Public Member Functions

- User (Robot _robot)
  The constructor.
- string GetMyself (bool show_privilege)
  Get the username who sent this HTTP request.
- void SetUser (string name, string password)
  Add a user or change the password for the existing user.
- void DeleteUser (string name)
  Delete a user account.
- string GetUser (bool show_privilege)
  Get the users list.
- void SetUserCheck (bool value)
  Enable or disable user authorization check.

Additional Inherited Members

7.43.1 Detailed Description

Manage user accounts.

7.43.2 Constructor & Destructor Documentation

7.43.2.1 Rovio.API.User.User (Robot *_robot*)

The constructor.

7.43.3 Member Function Documentation

7.43.3.1 void Rovio.API.User.DeleteUser (string *name*)

Delete a user account.

7.43.3.2 string Rovio.API.User.GetMyself (bool *show_privilege*)

Get the username who sent this HTTP request.

Todo Implement the proper return value: list of strings.

7.43.3.3 string Rovio.API.User.GetUser (bool *show_privilege*)

Get the users list.

Todo Implement the proper return value: list of strings.

7.43.3.4 void Rovio.API.User.SetUser (string *name,* string *password*)

Add a user or change the password for the existing user.

7.43.3.5 void Rovio.API.User.SetUserCheck (bool *value*)

Enable or disable user authorization check.

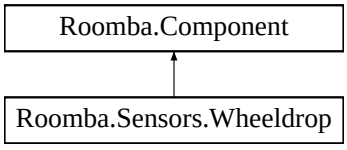
The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Rovio.cs

7.44 Roomba.Sensors.Wheeldrop Class Reference

The state of the Wheeldrop sensors: false = wheel up, true = wheel dropped.

Inheritance diagram for Roomba.Sensors.Wheeldrop:



Public Member Functions

- Wheeldrop (Robot _robot)

  The constructor.

Properties

- bool Right [get]

  Right wheeldrop sensor.
- bool Left [get]

  Left wheeldrop sensor.
- bool Caster [get]

  Caster wheeldrop sensor.

Additional Inherited Members

7.44.1 Detailed Description

The state of the Wheeldrop sensors: false = wheel up, true = wheel dropped.

Note: Some robots do not report the three wheel drops separately. Instead, if any of the three wheels drops, all three wheel-drop bits will be set. You can tell which kind of robot you have by examining the serial number inside the battery compartment. Wheel drops are separate only if there is an “E” in the serial number.

7.44.2 Constructor & Destructor Documentation

7.44.2.1 Roomba.Sensors.Wheeldrop.Wheeldrop (Robot *_robot*)

The constructor.

Parameters

_robot	

7.44.3 Property Documentation

7.44.3.1 bool Roomba.Sensors.Wheeldrop.Caster [get]

Caster wheeldrop sensor.

7.44.3.2 bool Roomba.Sensors.Wheeldrop.Left [get]

Left wheeldrop sensor.

7.44.3.3 bool Roomba.Sensors.Wheeldrop.Right [get]

Right wheeldrop sensor.

The documentation for this class was generated from the following file:

- C:/Users/Greg/Documents/GitHub/RoboLib/src/Roomba.cs

Index