

ur_variables Documentation

Generated by Doxygen 1.8.11

Contents

0.1	ur_variables Class Reference	1
0.1.1	Detailed Description	1
0.1.2	Constructor & Destructor Documentation	2
0.1.2.1	ur_variables()	2
0.1.3	Member Function Documentation	2
0.1.3.1	getAnalogInputStatus(int address)	2
0.1.3.2	getAnalogOutputCurrent(int address)	2
0.1.3.3	getAnalogOutputCurrentFactor(int address)	2
0.1.3.4	getAnalogOutputVoltage(int address)	3
0.1.3.5	getAnalogOutputVoltageFactor(int address)	3
0.1.3.6	getInput(int address)	3
0.1.3.7	getInputTool(int address)	3
0.1.3.8	getOutput(int address)	4
0.1.3.9	getOutputTool(int address)	4
0.1.3.10	getPayload()	4
0.1.3.11	isEStopped()	4
0.1.3.12	isMotionPossible()	5
0.1.3.13	isPowered()	5
0.1.3.14	setAnalogCurrent(int address, float current)	5
0.1.3.15	setAnalogCurrentFactor(int address, float factor)	5
0.1.3.16	setAnalogVoltage(int address, float voltage)	5
0.1.3.17	setAnalogVoltageFactor(int address, float factor)	6
0.1.3.18	setHigh(int address)	6
0.1.3.19	setLow(int address)	6
0.1.3.20	setOutput(int address, int state)	6
0.1.3.21	setOutputTool(int address, int state)	6
0.1.3.22	setPayload(float mass=0.900)	7
0.1.3.23	ur_variables_state_statusCallback(const industrial_msgs::RobotStatus::ConstPtr &msg)	7
0.1.3.24	ur_variables_statusCallback(const ur_msgs::IOStates::ConstPtr &msg)	7

0.1 ur_variables Class Reference

This class is used to manage the I/O screen of the UR3 robot using ros.

```
#include <ur_variables.h>
```

Public Member Functions

- [ur_variables](#) ()
- void [ur_variables_statusCallback](#) (const ur_msgs::IOStates::ConstPtr &msg)
- void [ur_variables_state_statusCallback](#) (const industrial_msgs::RobotStatus::ConstPtr &msg)
- void [setHigh](#) (int address)
- void [setLow](#) (int address)
- void [setPayload](#) (float mass=0.900)
- void [setOutput](#) (int address, int state)
- void [setOutputTool](#) (int address, int state)
- void [setAnalogCurrent](#) (int address, float current)
- void [setAnalogCurrentFactor](#) (int address, float factor)
- void [setAnalogVoltage](#) (int address, float voltage)
- void [setAnalogVoltageFactor](#) (int address, float factor)
- float [getPayload](#) ()
- int [getOutput](#) (int address)
- int [getInput](#) (int address)
- int [getOutputTool](#) (int address)
- int [getInputTool](#) (int address)
- float [getAnalogOutputCurrent](#) (int address)
- float [getAnalogOutputVoltage](#) (int address)
- float [getAnalogOutputCurrentFactor](#) (int address)
- float [getAnalogOutputVoltageFactor](#) (int address)
- float [getAnalogInputStatus](#) (int address)
- bool [isEStopped](#) ()
- bool [isPowered](#) ()
- bool [isMotionPossible](#) ()

0.1.1 Detailed Description

This class is used to manage the I/O screen of the UR3 robot using ros.

This class uses the ROS service SetIO and SetPayload to access and edit controller's data. You can modify the analog and digital outputs and read inputs and outputs status. This class also get the robot status (is the robot emergency stopped?, is the robot powered? is the robot ready to move?). You can also set current and or voltage to an analog output. Ypu require the ur_modern_driver to use this class. Unexpected behaviour could be notice when using current-voltage functions. To retrieve data a IOStatus subscriber is used in the class. This class doesn't depend on URScript.

Author

Barrero Lizarazo, Nicolas

Date

November 2018

0.1.2 Constructor & Destructor Documentation

0.1.2.1 `ur_variables::ur_variables ()`

The constructor initializes subscriptions to `io_states` and `robot_status` topics also creates two service clients; `set_io` and `set_payload`. No topic is advertised using this class.

0.1.3 Member Function Documentation

0.1.3.1 `float ur_variables::getAnalogInputStatus (int address)`

Gets analog input status in voltage or current according what is configured in the robot's controller.

Parameters

<i>address</i>	an int that represents analog inputs(0-1)
----------------	---

Returns

float status in V or mA according to robot's configuration.

0.1.3.2 `float ur_variables::getAnalogOutputCurrent (int address)`

Gets analog output current in mA. Unexpected behaviour can show up here.

Parameters

<i>address</i>	an int that represents analog outputs(0-1)
----------------	--

Returns

float current in mA (4mA-20mA)

0.1.3.3 `float ur_variables::getAnalogOutputCurrentFactor (int address)`

Gets analog output current as a percentage of max current(0->4mA, 1->20mA). Data is linear interpolated. Unexpected behaviour can show up here.

Parameters

<i>address</i>	an int that represents analog outputs(0-1)
----------------	--

Returns

float factor(0-1)

0.1.3.4 float ur_variables::getAnalogOutputVoltage (int *address*)

Gets analog output voltage in V. Unexpected behaviour can show up here.

Parameters

<i>address</i>	an int that represents analog outputs(0-1)
----------------	--

Returns

float voltage in V (0V-10V)

0.1.3.5 float ur_variables::getAnalogOutputVoltageFactor (int *address*)

Gets analog output voltage as a percentage of max voltage(0->0V, 1->10V). Data is linear interpolated. Unexpected behaviour can show up here.

Parameters

<i>address</i>	an int that represents analog outputs(0-1)
----------------	--

Returns

float factor(0-1)

0.1.3.6 int ur_variables::getInput (int *address*)

Gets a digital input state.

Parameters

<i>address</i>	an int that represents the digital input address(0-15)
----------------	--

Returns

state 1 for true/on 0 for false/off

0.1.3.7 int ur_variables::getInputTool (int *address*)

Gets a tool input state.

Parameters

<i>address</i>	an int that represents the tool input address(0-1)
----------------	--

Returns

state 1 for true/on 0 for false/off

0.1.3.8 int ur_variables::getOutput (int *address*)

Gets a digital output state.

Parameters

<i>address</i>	an int that represents the digital output address(0-15)
----------------	---

Returns

state 1 for true/on 0 for false/off

0.1.3.9 int ur_variables::getOutputTool (int *address*)

Gets a tool output state.

Parameters

<i>address</i>	an int that represents the tool output address(0-1)
----------------	---

Returns

state 1 for true/on 0 for false/off

0.1.3.10 float ur_variables::getPayload ()

Gets robot's payload.

Returns

payload mass (0-3)kg

0.1.3.11 bool ur_variables::isEStopped ()

Is the robot emergency stopped?

Returns

true or false

0.1.3.12 bool ur_variables::isMotionPossible ()

Is motion possible?

Returns

true or false

0.1.3.13 bool ur_variables::isPowered ()

Is the robot powered up?

Returns

true or false

0.1.3.14 void ur_variables::setAnalogCurrent (int *address*, float *current*)

Sets analog output current in mA. Unexpected behaviour can show up here.

Parameters

<i>address</i>	an int that represents analog outputs(0-1)
<i>float</i>	current in mA (4mA-20mA)

0.1.3.15 void ur_variables::setAnalogCurrentFactor (int *address*, float *factor*)

Sets analog output current as a percentage of max current(0->4mA, 1->20mA). Data is linear interpolated. Unexpected behaviour can show up here.

Parameters

<i>address</i>	an int that represents analog outputs(0-1)
<i>float</i>	factor(0-1)

0.1.3.16 void ur_variables::setAnalogVoltage (int *address*, float *voltage*)

Sets analog output voltage in V. Unexpected behaviour can show up here.

Parameters

<i>address</i>	an int that represents analog outputs(0-1)
<i>float</i>	voltage in V (0V-10V)

0.1.3.17 void ur_variables::setAnalogVoltageFactor (int *address*, float *factor*)

Sets analog output voltage as a percentage of max voltage(0->0V, 1->10V). Data is linear interpolated. Unexpected behaviour can show up here.

Parameters

<i>address</i>	an int that represents analog outputs(0-1)
<i>float</i>	factor(0-1)

0.1.3.18 void ur_variables::setHigh (int *address*)

Sets a digital output to true/1.

Parameters

<i>address</i>	an int that represents the address(0-15)
----------------	--

0.1.3.19 void ur_variables::setLow (int *address*)

Sets a digital output to false/0.

Parameters

<i>address</i>	an int that represents the address(0-15)
----------------	--

0.1.3.20 void ur_variables::setOutput (int *address*, int *state*)

Changes a digital output state.

Parameters

<i>address</i>	an int that represents the address(0-15)
<i>states</i>	an int 1 for true/on 0 for false/off

0.1.3.21 void ur_variables::setOutputTool (int *address*, int *state*)

Changes tool outputs state.

Parameters

<i>address</i>	an int that represents the address(0-1)
<i>states</i>	an int 1 for true/on 0 for false/off

0.1.3.22 void ur_variables::setPayload (float *mass* = 0 . 900)

Sets robot's payload. Default is 900g (current gripper)

Parameters

<i>mass</i>	float payload (0-3)kg
-------------	-----------------------

0.1.3.23 void ur_variables::ur_variables_state_statusCallback (const industrial_msgs::RobotStatus::ConstPtr & *msg*)

This is the regular Callback from a ros node, this function updates robot_Status data.

Parameters

<i>msg</i>	is the message type the nodes subscribes to: const industrial_msgs::RobotStatus::ConstPtr&
------------	--

0.1.3.24 void ur_variables::ur_variables_statusCallback (const ur_msgs::IOStates::ConstPtr & *msg*)

This is the regular Callback from a ros node, this function updates I/O data.

Parameters

<i>msg</i>	is the message type the nodes subscribes to: const ur_msgs::IOStates::ConstPtr&
------------	---

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

- /home/ctai/catkin_ws2/src/universal_robot/ur_variables/include/ur_variables/ur_variables.h

