RUTX10 Input/Output

The information in this page is updated in accordance with firmware version **RUTX_R_00.07.07.1**.

□

Contents

- 1 Summary
- 2 Physical characteristics and I/O pin mapping
  - 2.1 Power Socket Pinout
- 3 Status
  - 3.1 Status from command line
- 4 Post/Get
  - 4.1 Post/Get examples
    - 4.1.1 Setting
    - 4.1.2 Reading
- 5 Scheduler
- 6 I/O Juggler
  - 6.1 General
    - 6.1.1 Input
  - 6.2 Actions
    - 6.2.1 Type: Email
    - 6.2.2 Type: Output
    - 6.2.3 Type: HTTP
    - 6.2.4 Type: Script
    - 6.2.5 Type: Reboot
    - 6.2.6 Type: Profile
    - 6.2.7 Type: RMS
    - 6.2.8 Type: MQTT
    - 6.2.9 Type: WiFi
  - 6.3 Conditions
    - 6.3.1 Condition type: I/O
    - 6.3.2 Condition type: Boolean group
    - 6.3.3 Condition types: Date/Time

Summary

Inputs and **outputs** are used for monitoring and controlling a connected device or receiving signals from that device in order to trigger certain events.

This chapter of the user manual provides an overview of the Input/Output section for RUTX10 devices.

If you're having trouble finding this page or some of the parameters described here on your device's WebUI, you should **turn on "Advanced WebUI" mode**. You can do that by clicking the "Advanced" button, located at the top of the WebUI.

Physical characteristics and I/O pin mapping

Electrical characteristics and I/O pin mapping information are presented below.

Power Socket Pinout

- **Power**: 9-50 VDC positive (+).
- **Ground**: negative/ground (-).
- **Input**: digital non-isolated input.
  - *logical low level:* 0 - 5 V;
  - *logical high level:* 8 - 30 V.
- **Output**: digital open collector (OC) output; 30 V, 300 mA.

Status

The **Status** page displays the current states of the device's input and output pins:

You can invert an Input pin by clicking the on/off slider under the "Inversion" column or switch the state of an Output pin by clicking the on/off slider under the "State" column.

Status from command line

You can also obtain the status of input and output pins via the command line (CLI or SSH). List of possible ubus values in RUTX10 devices:

```
ioman.gpio.din1  - 4PIN input
ioman.gpio.dout1 - 4PIN output
```

Call command **ubus call value status** :

```
root@Teltonika-RUTX10:~# ubus call ioman.gpio.din1 status
{
        "value": "0",
        "direction": "in",
        "bi_dir": false,
```

```
        "invert_input": false
}
```

- **din1** is the input:
  - "value": "**0**" means low level state.
  - "value": "**1**" means high level state.
- **dout1** is the output:
  - "value": "**0**" means low level state.
  - "value": "**1**" means high level state.

Post/Get

Enabling **Post/Get** provides the possibility to control the state of an output via HTTP POST/GET requests. The figure below is an example of the Auth Settings section of the Post/Get page. It is used to turn Post/Get requests on or off and to set authentication parameters.

Field	Value	Description
Enable	off | on; default: **off**	Turns Post/Get on or off.
Access	io_state | io_type | io_value | io_invert; default: **io_state, io_type, io_value, io_invert**	Accessible methods. It is considered that all methods are allowed if this list is empty.
Username	string; default: **none**	Username used for authentication in POST/GET queries.
Password	string; default: **none**	Password used for authentication in POST/GET queries.
Confirm password	string; default: **none**	Repeat the password for confirmation.

Note: password fields are required to enable POST/GET funcionallity.

Post/Get examples

Use a web browser or any other compatible software to send HTTP POST/GET requests to the device.

Setting

Changing states of various I/O is possible using requests. Below is a table containing syntax examples of this usage:

Action	POST/GET URL
Turn Output state to high	http://192.168.1.1/cgi-bin/io_state?username=user1&password=user1&pin=dout1**&state=on**
Turn Output state to low after 5 seconds for 3 seconds	http://192.168.1.1/cgi-bin/io_state?username=user1&password=user1&pin=dout1&state=on**&delay=5&time=3**
Invert input	http://192.168.1.1/cgi-bin/io_invert?username=user1&password=user1&pin=din1**&invert=<0/1> (0 - disable inversion/1 - enable inversion**

Overview:

- 192.168.1.1 - device default LAN IP address; replace it in accordance with your own configuration.
- username - login name from Post/Get configuration.
- password- password from Post/Get configuration.
- state - turn Output on or off.
- invert - enables inversion on/off (only works with inputs).
- delay - defines a delay (in seconds) after which the specified action will be performed.
- time - defines a window of time during which the action will take place. For instance, if you post an *on* action while specifying *time=5*, the output will turn on and stay on for 5 seconds before turning off.

Delay and time parameters can be used together. For example, if delay is 10, time is 5, action is on, then 10 seconds after the execution of the command, the output will switch to *on* (or stay in *on* state if it was already that way), then after 5 more seconds it will switch to *off* state. In this case the overall command execution time is 15 seconds.

To use Post/Get via SSH instead of a browser, you may want to use the ***curl -X*** command. Simply add the same URL command in between quotes and specify the HTTP method.

Examples:

- **Switch output to High:**

```
curl -X GET
"http://192.168.1.1/cgi-bin/io_state?username=user1&password=user1&pin=dout1&
state=on"
```

- **Switch output to Low after delay:**

```
curl -X GET
"http://192.168.1.1/cgi-bin/io_state?username=user1&password=user1&pin=dout1&
state=off&delay=5"
```

Reading

Getting the current state of various I/O is possible using requests. Usage is very similar to the examples above:

Examples:

- **Read state of digital input/output:**

```
curl -X GET
"http://192.168.1.1/cgi-bin/io_value?username=user1&password=user1&pin=din1"
curl -X GET
"http://192.168.1.1/cgi-bin/io_value?username=user1&password=user1&pin=dout1"
```

Scheduler

The output **Scheduler** can be used to configure a timetable of when an output should be turned on or off, based on days of the week or month and hours of the day. The General Configuration section is used to turn the Output Scheduler on or off.

The Scheduler is configured in the form of **Instances**. A Scheduler Instance defines a time interval during which the state of an output associated with the instance will be set to "High". The Output Scheduler Instances list is empty by default. Click the 'Add' button in order to create a new Scheduler Instance:

After this you should be redirected to the configuration page for the newly added Instance which should look similar to this:

Field	Value	Description
Enable	off | on; default: **off**	Turns the Scheduler Instance on or off.
Pin	output pin; default: **Output (4)**	Output pin. The state of the selected output will be set to "High" during the time interval defined in the fields below.
Interval Type	Weekdays | Month Days; default: **Weekdays**	Selects the interval type for scheduler to use.
Start Day	[Monday..Sunday] | [1..31]; default: **Monday | 1**	The day that marks the start of the time interval.
Start Time	hh:mm ([00..23]:[00..59]); default: **12:00**	The hour and minute that mark the start of the time interval.
End Day	[Monday..Sunday] | [1..31]; default: **Tuesday | 1**	The day that marks the end of the time interval.
End Time	hh:mm ([00..23]:[00..59]); default: **12:00**	The hour and minute that mark the end of the time interval.
Force Last Day	off | on; default: **off**	Forces intervals to accept last day of month as a valid option if selected day does not exist during ongoing month. This field becomes visible only when 'Interval Type' is set to *Month Days*.

A Scheduler Instance will not work unless you turn both the the Scheduler service and the individual instance on:

I/O Juggler

The **I/O Juggler** is a feature that provides the possibility to create automated rules that perform certain actions based on Input state changes and other conditions. The operating sequence of I/O Juggler can be visualized as such:

When an Input Trigger occurs, the Input check to see if user-specified conditions are also met. If so, it executes a user-specified action.

For example, if we configure the I/O Juggler like this:

- **Trigger** - input state rising.
- **Condition** - 8:00 AM - 5:00 PM.
- **Action** - send email.

The operating sequence would look like this:

1. A connected device raises the input state.
2. This RUTX10 device checks whether the current time is between 8:00 AM and 5:00 PM.
    - **Yes** - RUTX10 sends an email.
    - **No** - RUTX10 doesn't send an email.

You can specify multiple actions and multiple conditions for any I/O Juggler rule. Selecting conditions is optional. If there are no set or configured I/O Juggler conditions, the operating sequence is as such:

General

The **General** section is used to Input Trigger rules, which can perform a specified user-configured action when the Input state changes and when certain other user-configured conditions are met (conditions are optional).

Before you can use the I/O Juggler, you must turn it on first (off by default). This can be done by setting the 'Enable' slider to "on":

Input

The **Input** is used to create Input Rules. An Input Rule is triggered by a user-specified input state change. When this trigger occurs, the device executes a specified command if certain specified

conditions are met (conditions are optional).

The Input list is empty by default. To create a new Input Rule, look to the Add Input section at the bottom of the page; select an input and click the 'Add' button:

After this you will be redirected to the configuration page for the newly added Rule, which should look similar to this:

Field	Value	Description
Enabled	off | on; default: **off**	Changes current profile to specified one when action triggers.
Trigger interval	integer [0..99999999]; default: **1**	The shortest amount of time (in seconds) between two triggers. If the input is triggered more than once in a time period shorter than the value specified in this field, the device will react to the first trigger.
Trigger	Rising | Falling | Both; default: **Rising**	Input state change that will trigger this rule.
Add actions	i/o juggler action(s); default: **none**	Actions that will be executed by this rule when the specified trigger and conditions (optional) occur.
Add conditions	i/o juggler conditions; default: **none**	Conditions that have to be met for the rule to take action. Conditions are optional.

Actions

The **Actions** section is used to manage and create new Actions that can be executed by the I/O Juggler Rules. The figure below is an example of the Actions section, which is empty by default.

To create a new Action, look to the Add an Action section at the bottom of the page; enter a custom name, select an action type and click the 'Add' button:

After this you will be redirected to the configuration page for the newly added Action, which should look similar to this (example for action 'Type: Reboot'):

Action configuration depends on the selected action 'Type'. Each Action can have one or multiple user-defined Conditions assigned to it. Actions that have Conditions assigned to them are executed only if the Conditions are met (a trigger occurrence is not enough to set off these actions). Assigning to Actions is optional.

You will find descriptions of each different action Type described on the sections below.

Type: Email

Sends an email to specified recipients. Requires an existing email account configuration on the device. Email accounts can be configured in the System → Administration → Recipients → Email Accounts page.

Field	Value	Description
Subject	string; default: **none**	Subject of email.
Text message	string; default: **none**	Email body text. Below this field you can find special codes that begin with the '%' sign. Each code represents a piece information related to the status of the device. Include these codes in the Text message for dynamic information reports.
Execution delay	positive integer; default: **none**	A period of time (in seconds) which has to pass after a trigger event before the action is executed.
Email account	email account; default: **none**	Email account used to send the email. Email accounts can be configured in the System → Administration → Recipients → Email Accounts page.
Recipient's email address	email; default: **none**	Email address(es) of the message's receiver(s).

Type: Output

Changes the state of a selected output pin.

Field	Value	Description
Execution delay	positive integer; default: **none**	A period of time (in seconds) which has to pass after a trigger event before this Action is executed.
Control	output; default: **Output (4)**	Selects the output controlled by this Action.
Revert	integer; default: **0**	After how many seconds the state will revert. If left as 0 or empty the state will not revert.
Maintain	off | on; default: **on**	When set to 'on', the output maintains its new state after reboot (the state is changed in the config file as well).
Invert	off | on; default: **off**	Inverts the state of the selected output, i.e., switches the output pin to a state that is opposite to its current one (high-to-low or low-to-high). This can be used instead of specifying a static state.
State copying	off | on; default: **off**	Copies the state from the selected input and applies it to the selected output. This can be used instead of specifying a static state.
State	High | Low; default: **High**	Specifies the state of the output pin that will be set by this Action.

Type: HTTP

Executes a HTTP POST/GET request.

Field	Value	Description
Method	Get | Post; default: **Get**	HTTP method to be used by this Action.
URL	string; default: **none**	URL to send the HTTP Post/Get request to.
Verify	off | on; default: **off**	Verifies the validity of certificates; only used with HTTPS.
Alt. parameters mode	off | on; default: **off**	Choose a different way to pass parameters. If method is POST parameters are passed in request body. If method is GET parameters are passed throught the URL.
Parameters	string; default: **none**	Parameters that are to be included the Post/Get request. Select a parameter (right drop-down box) and enter a custom name for it (left text box).
Text message	default: **Device name - %rn; Time stamp - %ts**	Message to send.
Custom headers	string; default: **none**	Allows to add custom headers to the HTTP requests.
Execution delay	positive integer; default: **none**	A period of time (in seconds) which has to pass after a trigger event before this Action is executed.

Type: Script

Executes a custom, user-written shell script.

Field	Value	Description
Execution delay	positive integer; default: **none**	A period of time (in seconds) which has to pass after a trigger event before this Action is executed.
Specify path	Upload a script |Specify path; default: **Upload a script**	Specifies whether the script should be uploaded from an external file or a path to an internal (on this device) script file should be specified.
Custom script | Script file	-(interactive button) | filepath	Shows file upload window on click or provides the possibility to specify a path to an internal script file. The name and selection type of this field depends on the value set in the 'Specify path' field.
Arguments	string; default: **none**	Optional arguments which can be provided for the script. You can include device information with values given below this input field.

Type: Reboot

Reboots the device.

Field	Value	Description
Execution delay	positive integer; default: **none**	A period of time (in seconds) which has to pass after a trigger event before this Action is executed.

Type: Profile

Switches to using a specified Configuration Profile.

Field	Value	Description
Execution delay	positive integer; default: **none**	A period of time (in seconds) which has to pass after a trigger event before this Action is executed.
Profile	profile; default: **default**	Configuration Profile that will be applied by this Action. Configuration Profiles can be defined in the System → Profiles page.

Type: RMS

Enables or disables RMS service.

Field	Value	Description
Execution delay	positive integer; default: **none**	A period of time (in seconds) which has to pass after a trigger event before this Action is executed.
Enable RMS	off | on; default: **off**	Specifies whether RMS service will be enabled or disabled with this action.

Type: MQTT

Executes a MQTT action.

Field	Value	Description

Text message	string; default: **none**	Below this field you can find special codes that begin with the '%' sign. Each code represents a piece information related to the status of the device. Include these codes in the Text message for dynamic information reports.
Topic	string; default: **none**	The name of the topic that the broker will subscribe to.
Client ID	string; default: **none**	Client ID to send with the data. If empty, a random client ID will be generated.
Hostname	host | ip; default: **none**	Broker's IP address or hostname.
Port	integer [0..65535]; default: **1883**	Broker's port number.
Keepalive	positive integer; default: **none**	The number of seconds after which the broker should send a PING message to the client if no other messages have been exchanged in that time
Qos	At most once (0) | At least once (1) | Exactly once (2); default: **At most once (0)**	A period of time (in seconds) which has to pass after a trigger event before this Action is executed.
Username	string; default: **none**	Username used for authentication to the Broker.
Password	string; default: **none**	Password used for authentication to the Broker.
TLS	off | on; default: **off**	Select to enable TLS encryption.
TLS Type	Certificate based | Pre-Shared-Key based; default: **Certificate based**	Type of TLS encryption.
Certificate based:Allow insecure connection	off | on; default: **off**	Allow not verifying server authenticity.
Certificate based:Certificate files from device	off | on; default: **off**	Choose this option if you want to select certificate files from device.
Certificate based:CA file	.crt file; default: **none**	Upload CA file.
Certificate based:Certificate file	.crt file; default: **none**	TLS client or server certificate file.
Certificate based:Key file	.key file; default: **none**	TLS client or server key file.
Pre-Shared-Key based:Pre-Shared-Key	string; default: **none**	The pre-shared-key in hex format with no leading “0x”.
Pre-Shared-Key based:Identity	string; default: **none**	Specify the Identity.

Type: WiFi

Sends an SMS message to specified recipients.

Field	Value	Description
Execution delay	positive integer; default: **none**	A period of time (in seconds) which has to pass after a trigger event before the action is executed.
Enable WiFi	off | on; default: **off**	Specifies whether all WiFi interfaces will be enabled/disabled with this action.

Conditions

The **Conditions** section is used to manage and created Conditions for I/O Juggler Actions. The figure below is an example of the Conditions section, which is empty by default.

To create a new Condition, look to the Add a Condition section; enter a custom name, select the Type of the Condition and click the 'Add' button.

After this you should be redirected to the configuration page of the newly added Condition. You will find descriptions of each different Condition type described on the sections below.

Condition type: I/O

I/O Condition type tracks the state of a selected input or output pin and considers the Condition as MET if that pin is in a user-specified state.

Field	Value	Description
I/O	inputs/outputs; default: **Output(4)**	Specifies the I/O pin to which this Condition is listening to.
State	High | Low; default: **High**	Specifies in what state the pin has to be in in order for the Condition to be met.

Condition type: Boolean group

Boolean Group will evaluate two or more existing conditions based on selected Boolean type. A Boolean Group condition will considered as MET based on the outcome of the evaluation performed along with one of the available Boolean types:

- **AND** - all selected conditions evaluate as TRUE.
- **NAND** - at least one selected condition evaluates as FALSE.
- **OR** - at least one selected condition evaluates as TRUE.
- **NOR** - none of the selected conditions evaluate as TRUE.

For example, if we have two time Conditions (hour and minute) and a third Condition of Boolean group: AND, both Conditions 1 & 2 have to be met for Condition 3 to be evaluated as TRUE.

In this case, if at least one condition is no longer met, the Boolean group: AND Condition is evaluated as FALSE.

Below is an example of Condition type: Boolean group configuration.

Field	Value	Description
Boolean type	AND | NAND | OR | NOR; default: **AND**	Type of boolean condition.
Add conditions	conditions; default: **none**	Specifies conditions which have to be met for action to occur.

Condition types: Date/Time

Lastly, you can configure Conditions based on date and time. Date/time Conditions can be configured based on:

- **Minute** of the hour
- **Hour** of the day
- **Day** of the **Week**
- **Day** of the **Month**
- **Day** of the **Year**

Each type can be configured in two distinct ways:

1. **Specific time** - configured as a single value which defines the exact time when the Condition is considered as TRUE. e.g.:
    - Hour: **8**
    - Day of Month: **1**
2. **Time interval** - configured with a start and an end value which define a window of time during which the Condition is considered as TRUE. e.g.:
    - Hours: **8-12**
    - Days of Month: **1-12**