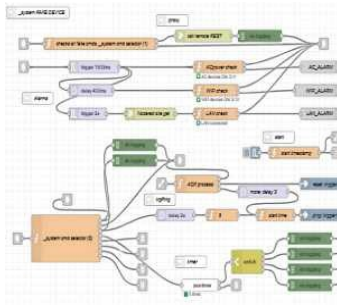


## tuyaDAEMON device \_system



Model: ver 2.2.0

Tuya: custom

Tuya Mod: SW only device

SYSTEM utilities and control

This is a fake SW device, to extend the capabilities of CORE  
Requires: CORE. Areas: security and alarms, benchmarks, DBs, timers, OO, network

Power supply:

Broadcast:

Capabilities: **SET,GET,SCHEMA**

Reference:

Infos: [https://github.com/msillano/tuyaDAEMON/wiki/custom-device-\\_system](https://github.com/msillano/tuyaDAEMON/wiki/custom-device-_system)

Sellers: <https://github.com/msillano/tuyaDAEMON>

### User defined Data Points:

<b>_ACpower</b>		RO		true false	
	PUSHed when all AC devices are down. GET: last value				
<b>_ACunconnected</b>		RO		a list like: ['dev_name1', 'dev_name2'...]	
	List of all AC devices unconnected. From tuya-smart-device-node. PUSHed at any change.				
<b>_beep</b>		WO		any	
	SET: trigger, sends a short beep Requires node-red UI visible				
<b>_beep_loop</b>		SKIP		{'count': N, 'timeout': XXXX [ms]}	
	SET: trigger, starts a beep repetition. tuyaDEAMON-chain. See <a href="https://github.com/msillano/tuyaDAEMON/wiki/tuyaDAEMON-as-event-processor#iteration">https://github.com/msillano/tuyaDAEMON/wiki/tuyaDAEMON-as-event-processor#iteration</a>				
<b>_benchmark</b>		RW		a task:{ device, property (, value (,timeout)) }	
	Defines the task used to run the benchmarks. Defaults: _system._zeroTask, timeout: 10000 ms				
<b>_benchmark_end</b>		TRG		any	
	To terminate a running benchmark Internal use only				
<b>_benchmark_step</b>		TRG		any	
	Called via share by the task when running benchmark Internal use only				
<b>_doBenchmark</b>		RW		any	
	SET: trigger, starts a benchmark, as defined by _benchmark				
<b>_doSCHEMA</b>		RW	string	device	
	SET: trigger, does a GET for all available DPs of a device Return the list of updated PDs. GET: last list.				
<b>_doTrigger</b>		WO	int	trigger value	
	Help property to send a TRIGGER to tuya. Requires TRIGGER module.				



<b>_doUP DATE</b>		RW	string	'real' 'virtual' 'fake' null (=all)	
	SET trigger: executes doSCHEMA for every local connected device Return the list of updates devices. GET: last list				
<b>_exec</b>		RW			
	To do an O.S. task. Waits until task ends GET: Returns last output				
<b>_LANn et</b>		RO	boolean	true false	
	Signals when the WEB access is down. See also node LAN_ALARM. PUSHed at any change.				
<b>_lastst art</b>		RO	string	'2021-03-24 16:06:33'	
	PUSHed some seconds after the restart. See also node start_DAEMON				
<b>_play</b>		WO	see note	string buffer	
	String: text to speech. Buffer WAV: play sound. Choose the voice in the 'play audio' node				
<b>_proxy</b>		RW		std_msg: { (remote:'name',) (device:'name pseudoDP id' (, property:'name dp' (, value:'any'))))}	
	SET: uses REST to connect a remote tuyadaEMON. GET: returns last response from 'remote', in global.alldevices. <system>. <proxy>				
<b>_sqlD Blocal</b>		RW	string	sql	
	SET: send to local DB a sql (value) and log the response GET: get the last DB response				
<b>_sqlD Bremote</b>		RW	string	sql	
	Default: 0,1,2 DB. CUSTOMIZABLE Same as sqlDBlocal:				
<b>_timer List</b>		RO		[{id:<any>, datetime: N, dateTimeStr: 'xxx', alarmPayload: {<share>}},...]	
	GET a list of all active timers. see <a href="https://flows.nodered.org/node/node-red-contrib-jsontimer">https://flows.nodered.org/node/node-red-contrib-jsontimer</a> for more info				
<b>_timer OFF</b>		WO	string	the timer 'id'	
	To delete a timer. The 'id' is mandatory. see <a href="https://flows.nodered.org/node/node-red-contrib-jsontimer">https://flows.nodered.org/node/node-red-contrib-jsontimer</a> for more info				
<b>_timer ON</b>		WO		{('id': '<any>'), 'timeout datetime time': <atime>, 'alarmPayload': { 'share': [... ] } }	
	Set a new timer and the 'share' actions associated. see <a href="https://flows.nodered.org/node/node-red-contrib-jsontimer">https://flows.nodered.org/node/node-red-contrib-jsontimer</a> and <a href="https://github.com/msillano/tuyaDAEMON/wiki/tuyaDAEMON-global.alldevices#share-actions">https://github.com/msillano/tuyaDAEMON/wiki/tuyaDAEMON-global.alldevices#share-actions</a>				
<b>_toDe bug</b>		WO		<string> or {[<string>,<object>,...]}	
	Sends the value to debug pad.				
<b>_toFas tIN</b>		WO		std_msg: { (remote:'name',) (device:'name id' (, property:'name dp' (, value:'any'))))}	
	To inject a msg to 'fast_cmds' node. Format: see core.'fast_cmds' node.description				
<b>_toLog ging</b>		WO		like std. answer: 'payload':{ 'deviceId': <id gateway-id>, 'data': { ('t': <timestamp>,) ('cid': <cid>,) 'dps': { <dp>:<value>}}};	
	To process a response: echo on Pad, DB and global.tuyastatus updated. Format: see core.'logging' node.description				

<b>_toLowIN</b>		WO		{'toDev':<deviceID>, 'payload':{<for the tya_smart_device>}}.	
	To inject a msg to a real device, using the 'low_level_IN' node To test new features (in devices or connection node). see core.'low_level_IN' node.description				
<b>_toShare</b>		WO		{ 'share': [ { 'test': [ '<test1>',...], 'action': [{ 'device': 'name id', 'property': 'name id @exp', 'value': 'any @exp' },... ] },... ] }	
	To inject a msg to 'share IN' node. see 'share': <a href="https://github.com/msillano/tuyaDAEMON/wiki/tuyaDAEMON-global.alldevices#share-actions">https://github.com/msillano/tuyaDAEMON/wiki/tuyaDAEMON-global.alldevices#share-actions</a>				
<b>_toStdCmd</b>		WO		std_msg: { remote:'name', device:'name id', property:'name dp', value:'any' }	
	To inject a msg to 'std_cmd' node. Format: see core.'std_cmd' node.description				
<b>_toWarn</b>		WO	string	any	
	Show in locale the value using node.warn().				
<b>_tuyasatus</b>		WO		like a std_msg: { ('device':<name>, ('property':<name>, ('value':<any>))) }	
	Alternative access to global.tuyastatus (e.g. from remote). Performs: LIST, if value:undef, SCHEMA, if value.property:undef, GET, if .value.value: undef, else SET				
<b>_WiFinet</b>		RO		true false	
	Signals when all WIFI devices are down. From tuya-smart-device-node. See also node WiFi_ALARM. PUSHed at any change.				
<b>_WiFiuconnected</b>		RO		a list like: ['dev_name1', 'dev_name2'...]	
	List of all WiFi devices unconnected. From tuya-smart-device-node. PUSHed at any change.				
<b>_zerolog</b>		RW		User free to set any DPtype and DPdecode.	
	Task for benchmark with log (debug-pad,DB, MQTT). Returns the 'value' normalized/encoded as defined (for tests).				
<b>_zerotask</b>		RW		any	
	Task for benchmark without log. Triggers the fastest fake task: it does nothing.				