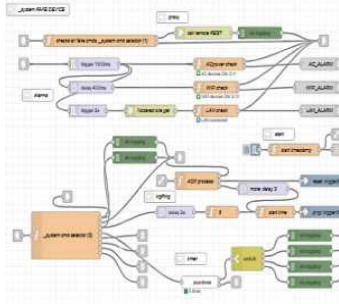


## *tuyaDAEMON device\_system*



Model: ver 2.2.2

Tuya: custom

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: **BAT**

Broadcast: **WiFi**

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	
	<i>PUSHed when all AC devices are down. GET: last value</i>				
<b>_ACunconnected</b>		RO		a list like: ['dev_name1', 'dev_name2'...]	
	<i>List of all AC devices unconnected. From tuya-smart-device-node. PUSHed at any change.</i>				
<b>_beep</b>		WO		any	
	<i>SET: trigger, sends a short beep Requires node-red UI visible</i>				
<b>_beep_loop</b>		SKIP		{'count': N , 'timeout': XXXX [ms]}	
	<i>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></i>				
<b>_benchmark</b>		RW		a task:{ device, property (, value (,timeout)) }	
	<i>Defines the task used to run the benchmarks, must have a 'share' (see _zerotask) Defaults: _system._zeroTask, timeout: 10000 ms</i>				
<b>_benchmark_end</b>		TRG		any	
	<i>To terminate a running benchmark Internal use only</i>				
<b>_benchmark_step</b>		TRG		any	
	<i>Called via share by the task when running benchmark Internal use only</i>				
<b>_doBenchmark</b>		RW		any	
	<i>SET: trigger, starts a benckmark, as defined by _benchmark</i>				
<b>_doSCHEMA</b>		RW	string	device	
	<i>SET: trigger, does a GET for all available DPs of a device Return the list of udated PDs. GET: last list.</i>				
<b>_doTrigger</b>		WO	int	trigger value	
	<i>Help property to send a TRIGGER to tuya. Requires TRIGGER module.</i>				
<b>_doUPDATE</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			
_exec		RW		string (command line)
	To do an O.S. task.SET waits until task ends, Returns output GET: Returns last output			
_LANnet		RO	boolean	true false
	Signals when the WEB access is down. See also node LAN_ALARM. PUSHed at any change.			
_laststart		RO	string	'2021-03-24 16:06:33'
	PUSHed some seconds after the restart. See also node start_DAEMON			
_play		WO	see note	string buffer
	String: text to spreach. Buffer WAV: play sound. Choose the voice in the 'play audio' node			
_proxy		RW		std_msg: { (remote:'name',) (device:'name pseudoDP id' (, property:'name dp' (, value:'any'))))}
	SET: uses REST to connect a remote tuyaDAEMON. GET: retuns last response from 'remote', in global.alldevices.<system>.<proxy>			
_sqlDBlocal		RW	string	sql
	SET: send to local DB a sql (value) and log the response GET: get the last DB response			
_sqlDBremote		RW	string	sql
	Default: 0,1,2 DB. CUSTOMIZABLE Same as sqlDBlocal:			
_timerList		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			
_timerOFF		RW	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			
_timerON		RW		{('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>			
_toDebug		WO		<string> or {[<string>,<object>,...]}
	Sends the value to debug pad.			
_toFastIN		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			
_toGlobal		WO		{variable:any, value: value NULL}
	Alternative access to global (e.g. from remote). if value.value = NULL => GET global.any, else SET			
_toLogging		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>_tuyastatus</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>_WiFiunconnected</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 DPTtype 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.				