hackathon21-vkeypad

Jose Luis Aracil

Nov 25, 2021

CONTENTS:

CHAPTER

ONE

MODULES

1.1 conin

1.1.1 conin.cmd.seq

Description: [**list**] Process a key sequence (list of objects) or single object

- **“t”**: [**string**] Action type p = push, h = hold, r = release, w = wait
- **“c”**: [**string**] [**optional**] Key code. Mandatory for p, h and r action types
- **“on”**: [**number**] [**optional**] Time ON in milliseconds for p action type. Default 100ms
- **“off”**: [**number**] [**optional**] Time OFF in milliseconds for p action type. Default 100ms
- **“r”**: [**number**] [**optional**] Number of keystrokes for p action type. Default 1
- **“ms”**: [**number**] [**optional**] Wait milliseconds for w action type. Default 200ms

Returns: [**nil**]

1.1.2 conin.cmd.alive

Description: [**object**] Refresh keep alive timer

- **“ms”**: [**number**] [**optional**] Keep alive timeout in milliseconds. Default 10000ms

Returns: [**nil**]

1.1.3 conin.evt.key

Description: [**object**] Key press/release event

- **“c”**: [**string**] Key code
- **“st”**: [**boolean**] Key status true = pressed, false = released

1.1.4 conin.stat.alive

Description: [**boolean**] Alive status true = alive, false = dead

1.2 event

1.2.1 event.cmd.set

Description: [**object**] Register a new event. The event object must have at least the id property but you can add the properties you need to describe the event

- **"id"**: [**string**] object identifier

Returns: [**nil**]

1.2.2 event.cmd.get

Description: [**object**] Get events from events ring buffer

- **"bc"**: [**number**] Boot counter of the last readed event
- **"ec"**: [**number**] Event counter of the last readed event

Returns: [**list**] list of events that have occurred after boot counter and event counter

1.2.3 event.cmd.last

Description: [**nil**] Returns the boot counter value (bc), and the event couter value (ec)

Returns: [**object**] Returns the boot counter and event counter value.

- **"bc"**: [**number**] boot counter of the last readed event
- **"ec"**: [**number**] event counter of the last readed event

1.2.4 event.cmd.config

Description: [**object**] Set/Get event module configuration

- **"max_event_entries"**: [**number**] [**optional**] Maximum number of events kept in RAM
- **"sync_delay"**: [**number**] [**optional**] Normal events notification delay in milliseconds
- **"prio_sync_delay"**: [**number**] [**optional**] High priority events notification delay in milliseconds
- **"notify"**: [**object**] [**optional**] events to be notified eg. {"system_boot":1, "some_low_prio_event":0} use 1 for hihg priority events 0 otherwise

Returns: [**object**] returns event module configuration.

- **"max_event_entries"**: [**number**] Maximum number of events kept in RAM
- **"sync_delay"**: [**number**] Normal events notification delay in milliseconds
- **"prio_sync_delay"**: [**number**] High priority events notification delay in milliseconds
- **"notify"**: [**object**] events to be notified eg. {"system_boot":1, "some_low_prio_evt":0} 1 = hihg priority 0 = low priority

1.3 io

1.3.1 io.cmd.set_out

Description: [**number**] Puts outputs to [on] state. Each bit represents one output. E.g. value 3 enables outputs 0 and 1

Returns: [**number**] Returns current outputs status

1.3.2 io.cmd.clr_out

Description: [**number**] Puts outputs to [off] state. Each bit represents one output. E.g. value 3 disables outputs 0 and 1

Returns: [**number**] Returns current outputs status

1.3.3 io.cmd.put_out

Description: [**object**] Puts outputs.

- **"set"**: [**number**] [**optional**] Puts outputs to [on] state. Each bit represents one output. E.g. value 3 enables outputs 0 and 1
- **"clr"**: [**number**] [**optional**] Puts outputs to [off] state. Each bit represents one output. E.g. value 3 disables outputs 0 and 1

Returns: [**number**] Returns current outputs status

1.3.4 io.cmd.get_inp

Description: [**nil**] Polls inputs status

Returns: [**number**] Returns current inputs status. Each bit represents one input. E.g. value 3 represents inputs 0 and 1 in [on] state

1.3.5 io.stat.inputs

Description: [**number**] State of each input. Each bit represents an input.

1.3.6 io.stat.outputs

Description: [**number**] State of each output. Each bit represents an output.

1.4 javascript

This module lets you implement some logic via javascript. You can embed a single file via its menuconfig parameter, and it will be loaded when the module starts

1.4.1 Available Javascript DOM Functions

```
message = { "topic": string, "rtopic": string, "value": any }
error = { "error": true, "id": number, "desc": string }
options = { "sticky": true, "non_recursive": true, "integer": true }

pubsub.subscribe(topic, function(message)) -> handler
pubsub.unsubscribe(handler)

pubsub.publish(topic, value [, options])
pubsub.call(topic, value, function(message), function(error), timeout_ms [, options])
```

Options are *optional*, and there is no need to specify them all.

- **sticky**: The pubsub will store the message value and will send it to future subscriptions on this topic. The next publish to the same topic will modify (if it's also sticky) or remove the stored value
- **non_recursive**: Prevents a message from traversing up the pubsub tree. Eg. *test* subscription will not receive a non_recursive *test.hello* publish
- **integer**: Needed since javascript lacks an integer type which a C subscriber might be expecting

1.4.2 Caveats

- If a script blocks for too long, it will be killed and **wont** be restarted to avoid a soft-lock situation
- Right now, this module uses a single subscriptor for all the javascript subscriptions on the C side, so there might be undesired callbacks calls if there are overlapping subscriptions

  When subscribed to *test* and *test.hello*, a published message to *test.hello* will match **both subscriptions**.

  This will cause two messages to be sent to the javascript side, with the topic *test.hello* which once again will match both subscriptions on the javascript side, and will run **both callbacks two times**.
- The following module commands to upload a new script are only available to aid the development process, but nothing will be stored on the board and it will be lost on restart

1.4.3 Example

```
pubsub.subscribe("test", function (m) {
    console.log("Received a message on", m.topic, ":", m.value)
})

pubsub.publish("test.int", 42, { "integer": true })
```

1.4.3.1 javascript.cmd.load

Description: [**string**] Reinitialize the javascript engine, and load the published script. This command will accept a string or a buffer

Returns: [**nil**] Ok

1.4.3.2 javascript.cmd.eval

Description: [**string**] Load the published script without reinitializing the engine. This command will accept a string or a buffer

Returns: [**nil**] Ok

1.4.3.3 javascript.cmd.upload

Description: [**object**] Begin a chunked script upload

- **"size"**: [**number**] Total size in bytes
- **"sha256"**: [**bin**] [**optional**] sha256 of the completed file. Wont check the hash if omitted
- **"persist"**: [**boolean**] [**optional**] store the script in flash

Returns: [**nil**] Ok

1.4.3.4 javascript.cmd.upload.chunk

Description: [**bin**] Upload a script chunk

Returns: [**nil**] Ok

1.4.3.5 javascript.evt.stdout

Description: [**string**] Standard output of the javascript engine

1.4.3.6 javascript.evt.stderr

Description: [**string**] Error output of the javascript engine

1.5 lock

1.6 log

1.7 session

1.7.1 ses.cmd.ping

Description: [**nil**] Ping command

Returns: [**nil**]

1.8 storage

1.9 stream

1.10 sys

1.10.1 sys.cmd.info

Description: [**nil**] Retrieve hardware and runtime device information

Returns: [**object**]

- **"product"**: [**string**] Product Name
- **"board"**: [**string**] Board Name
- **"firmware_version"**: [**number**] Firmware version
- **"uptime"**: [**number**] Time since last boot in us.
- **"clock"**: [**number**] Current time, in ms since Unix EPOCH
- **"name"**: [**string**] Device name

1.10.2 sys.cmd.reboot

Description: [**object**] Reboot the device.

- **"now"**: [**boolean**] [**optional**] if true, forces a reboot, without taking into consideration the reboot locks. If false, waits until all the locks are given. Default to false.

Returns: [**nil**]

1.10.3 sys.cmd.mac_info

Description: [**nil**] Returns the MAC address of all the device's interfaces.

Returns: [**object**]

1.10.4 sys.cmd.time

Description: [**object**] Sets the device time of day

- **"time"**: [**number**] Number of seconds since midnight.
- **"tz_data"**: [**string**] [**optional**] Timezone of the "time" parameter. If not set, the timezone does not change. Default = UTC

Returns: [**object**]

- **"time"**: [**number**] Number of seconds since midnight.
- **"tz_data"**: [**string**] Timezone of the "time" parameter.

1.10.5 sys.cmd.name

Description: [**object**] Sets the device name

- **"name"**: [**string**] Name of the device

Returns: [**object**]

- **"name"**: [**string**] Name of the device

1.10.6 sys.cmd.netmanager

Description: [**object**] Configure the device network manager

- **"drv"**: [**string**] Name of the module which will manage the network

Returns: [**object**]

- **"drv"**: [**string**] Name of the module which will manage the network

1.10.7 sys.cmd.netmanager.restart

Description: [**nil**] Requests the device network manager to restart the network

1.10.8 sys.stat.name

Description: [**nil**] State returning the device name

Returns: [**string**] Name of the device

1.10.9 sys.stat.netmanager.ip

Description: [**nil**] State returning the current network ip or NIL if there is no active connection

Returns: [**object**]

- **"ip"**: [**string**] IP Address
- **"netmask"**: [**string**] Network Mask
- **"gw"**: [**string**] Gateway

1.10.10 sys.stat.netmanager.drv

Description: [**nil**] State returning the current device network manager

Returns: [**string**] Name of the module managing the network

1.10.11 sys.stat.discoverable

Description: [**nil**] State containing if the device is in bluetooth's discover mode or not.

Returns: [**boolean**] True if device is discoverable

1.10.12 sys.stat.time.synced

Description: [**nil**] State containing if the device has a correct time, either from an external call to "sys.cmd.time" (usually from the cloud) or a startup from an internal RTC already setup.

Returns: [**boolean**] True if device is discoverable

1.10.13 sys.evt.shutdown

Description: [**object**] Event emitted when the device begins the shutdown procedure. This event is emmited independently of the reboot tipe (forced or gracefull)

- **"time_left"**: [**number**] time left in ms to reboot. Default is 5000 (5 seconds).

1.11 tcp

1.12 ui

This module provides a basic user interface such as led signaling, buzzer, discover button, ...

1.12.1 ui.cmd.beep

Description: [**object**] produces sounds in the buzzer

- **"on"**: [**number**] [**optional**] Set buzzer on [n/10] seconds
- **"off"**: [**number**] [**optional**] Set buzzer off [n/10] seconds after time on
- **"repeat"**: [**number**] [**optional**] Repeat [n] times on/off sequence

Returns: [**nil**]

1.13 wifi

1.13.1 wifi.cmd.config

Description: [**object**] configures wifi connection

- **"ssid"**: [**string**] [**optional**] wifi name [ssid]
- **"pass"**: [**string**] [**optional**] wifi password
- **"disable"**: [**boolean**] [**optional**] disables wifi if is set to true

Returns: [**object**] Returns an object with current settings {ssid, pass, disable}

Note: Send an empty object to get current settings

1.13.2 wifi.cmd.info

Description: [**nil**] gets wifi information

Returns: [**object**]

- **"status"**: [**string**] current wifi status [Invalid, Down, Disconnected, Connected, Disconnecting, Connecting]
- **"ip"**: [**string**] current wifi interface IP
- **"ssid"**: [**string**] current wifi ssid
- **"disable"**: [**string**] true if wifi is disabled

1.13.3 wifi.cmd.scan

Description: [**nil**] performs a wifi scan

Returns: [**list**] returns a list of APs found.

- **"ssid"**: [**string**] AP ssid
- **"bssid"**: [**string**] AP bssid
- **"mode"**: [**string**] AP mode [OPEN, WEP, WPA_PSK, WPA2_PSK, WPA_WPA2_PSK, WPA2_ENTERPRISE]
- **"chan"**: [**number**] AP channel
- **"rssi"**: [**number**] AP signal strength
- **"wps"**: [**boolean**] AP WPS support

Warning: Number of results are limited to 20

1.13.4 wifi.cmd.restart

Description: [**nil**] performs a wifi restart

Returns: [**nil**]

1.13.5 wifi.stat.link

Description: [**number**] status of wifi link [Invalid = -1, Down = 0, Disconnected = 1, Connected = 2, Disconnecting = 3, Connecting = 4]