

API pilight

Function	Params	Return
isRunningReceive		int(result) : state of receiver, 1 if running, 0 if not running, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
startReceive		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
isRunningServer		int(result) : 1 if running, 0 if not running, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
send	str(protocol) : name of protocole str(unit) : unit value str(id) : id of device str(value) : value to apply	int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
startServer		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
stopServer		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message

API i2c_analog

Function	Params	Return
read	int(pin) : value between 0 and 9	int(result) : value, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
readBattery		int(error) : 0 if no error, 1 if error str(message) : optional message array(result) : [level] [pourcent] [life] [lifetime], -1 if error
readAll		int(error) : 0 if no error, 1 if error str(message) : optional message array(result) : array of 10 values, -1 if error

API i2c_digital

Function	Params	Return
rvb	str(color) : optional, value could be 'black' 'red' 'green' 'blue' 'pink' 'yellow' 'cyan' or 'white', default is black int(pin_blue) : value between 0 and 15 int(reverse) : optional, 0 or 1, default is 0 int(pin_red) : value between 0 and 15 int(pin_green) : value between 0 and 15 int(autoOutput) : optional, 1 for auto-set as output, 0 if not, default is 1	int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message

pwmStopAll		int(result): 1 if success, -1 if error int(error): 0 if no error, 1 if error str(message): optional message
pwmDutyStart	int(dutyCycle): optional, value between 0 and 100 in pourcent, default is 50% int(pin): value between 0 and 15 int(frequency): optional, value between 0 and 200 in hertz, default is 100hz int(reverseCycle): optional, 0 or 1, default is 0	int(result): 1 if success, -1 if error int(error): 0 if no error, 1 if error str(message): optional message
toggleOutput	int(pin): value between 0 and 15 int(autoOutput): optional, 1 for auto-set as output, 0 if not, default is 1	int(result): 1 if success, 0 if not, -1 if error int(error): 0 if no error, 1 if error str(message): optional message
readSonar	int(pinTrig): value between 0 and 15 int(pinEcho): value between 0 and 15	int(result): distance, -1 if error int(error): 0 if no error, 1 if error str(message): optional message
pwmIsRunning	int(pin): value between 0 and 15	int(result): 1 if running, 0 if not running, -1 if error int(error): 0 if no error, 1 if error str(message): optional message
pwmChangeDuty	int(dutyCycle): optional, value between 0 and 100 in pourcent, default is 50% int(pin): value between 0 and 15	int(result): 1 if success, -1 if error int(error): 0 if no error, 1 if error str(message): optional message
pwmStop	int(pin): value between 0 and 15	int(result): 1 if success, -1 if error int(error): 0 if no error, 1 if error str(message): optional message
outputAll	int(value): 0 for low, 1 for high	int(error): 0 if no error, 1 if error str(message): optional message array(result): array of all input, -1 if error
isInput	int(pin): value between 0 and 15	int(result): 1 if success, 0 if not, -1 if error int(error): 0 if no error, 1 if error str(message): optional message
setInput	int(pin): value between 0 and 15	int(result): 1 if success, -1 if error int(error): 0 if no error, 1 if error str(message): optional message
setOutput	int(pin): value between 0 and 15	int(result): 1 if success, -1 if error int(error): 0 if no error, 1 if error str(message): optional message
output	int(pin): value between 0 and 15 int(value): 1 for high, 0 for low int(autoOutput): optional, 1 for auto-set as output, 0 if not, default is 1	int(result): 1 if success, -1 if error int(error): 0 if no error, 1 if error str(message): optional message
pullup	int(pin): value between 0 and 15 int(enable): 0 for single input pin, 1 for pull-up resistor	int(result): 1 if success, -1 if error int(error): 0 if no error, 1 if error

		str(message) : optional message
input	int(pin) : value between 0 and 15 int(autoInput) : optional, 1 for auto-set as input, 0 if not, default is 1	int(result) : value of pin, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
pwmBlinkStart	int(pin) : value between 0 and 15 int(frequency) : optional, value between 0 and 200 in hertz, default is 100hz float(dutyCycle) : optional, value between 0.0 and 1.0, default is 1 int(reverseCycle) : optional, 0 or 1, default is 0	int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
config	int(pin) : value between 0 and 15 int(mode) : 0 for output, 1 for input	int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
inputAll		int(error) : 0 if no error, 1 if error str(message) : optional message array(result) : array of all input, -1 if error
isOutput	int(pin) : value between 0 and 15	int(result) : 1 if success, 0 if not, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message

API espeak

Function	Params	Return
setRate	str(value) : optional, default is 200	int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
getVolume		int(result) : volume in pourcent, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
isSpeaking		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
setVoice	str(value) : optional, default is 200	int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
getVoice		int(error) : 0 if no error, 1 if error str(message) : optional message str(result) : voice, -1 if error
speak	int(rate) : optional, default is 200 str(voice) : optional, else use default str(text) : string to speak int(volume) : optional, default is 100 pourcent	int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
getVoices		int(error) : 0 if no error, 1 if error

		str(message) : optional message array(result) : list of voices, -1 if error
getRate		int(result) : value, -1 if error int(error) : 0 if no error; 1 if error str(message) : optional message
setVolume	int(value) : optional, default is 100 percent	int(result) : 1 if success, -1 if error int(error) : 0 if no error; 1 if error str(message) : optional message

API wiimote

Function	Params	Return
setLed	int(value) : mask number of leds to enable, number between 0 and 8	int(result) : 1 if success, 0 if not running, -1 if error int(error) : 0 if no error; 1 if error str(message) : optional message
isRunning		int(result) : 1 if running, 0 if not, -1 if error
read	int(subValue) : optional int(value) : optional	int(error) : 0 if no error; 1 if error str(message) : optional message array(result) : array of results, -1 if error
stop		int(result) : 1 if success, 0 if not running, -1 if error int(error) : 0 if no error; 1 if error str(message) : optional message
start	int(wm_nunchuk) : optional, can be 0 or 1, default is 0 int(wm_motionplus) : optional, can be 0 or 1, default is 0 int(wm_ir) : optional, can be 0 or 1, default is 0 int(wm_acc) : optional, can be 0 or 1, default is 0	int(result) : 1 if success, 0 if timeout, -1 if error int(error) : 0 if no error; 1 if error str(message) : optional message
setRumble	float(value) : optional, default is 0.25	int(result) : 1 if success, 0 if not running, -1 if error int(error) : 0 if no error; 1 if error str(message) : optional message

API network

Function	Params	Return
shareMountAll		int(result) : 1 if success, -1 if error int(error) : 0 if no error; 1 if error str(message) : optional message
shareUmount	str(name) : name of mount folder to umount	int(result) : 1 if success, 0 if failed, -1 if error int(error) : 0 if no error; 1 if error str(message) : optional message
getHostname		int(error) : 0 if no error; 1 if error str(message) : optional message str(result) : [str(name)] [str(ip)], -1 if error
shareMount	str(login) : optional share username str(password) : optional share password str(path) : path of folder like data/Torrents	int(result) : 1 if success, -1 if error int(error) : 0 if no error; 1 if error str(message) : optional message

	str(name) : name of mount folder to create/use str(host) : host address like 192.168.1.254	
setHostname	str(name) : new hostname to apply	int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
shareUmountAll		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
shareGetCategory	str(category) : must be "video" or "audio"	int(error) : 0 if no error, 1 if error str(message) : optional message array(result) : [str(category)] [int(count)] [array(results)], -1 if error
ping	str(url) : optional, default is "www.google.com"	int(result) : ping value, 0 if error int(error) : 0 if no error, 1 if error str(message) : optional message

API script

Function	Params	Return
start	str(filename) : name of script to start	int(result) : 1 if success, 0 if already started, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
running		int(result) : list of running scripts, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
stop	str(filename) : name of script to start	int(result) : 1 if success, 0 if already stopped, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
stopAll		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
isRunning	str(filename) : name of script to start	int(result) : 1 if running, 0 if not running, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message

API mediaplayer

Function	Params	Return
getVolume		int(result) : volume in percent, -1 if error
seekForward600		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
speedBackward		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message

setVolumeDown		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
seekBack600		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
getMedia		str(result) : name of playing media
toggleSubtitles		int(result) : 1 if enable, 0 if disable, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
getPosition		int(result) : position in secondes
getScreenInfo		str(result) : width x height, of screen
play	str(options) : optional, see omxplayer options str(audioOutput) : optional, can be "hdmi" or "local" str(filename) : optional, filename for display str(filepath) : full filepath of media str(videoRatio) : optional, can be "stretch" or "zoom" or "pan" int(volume) : optional, default is 50 pourcents	int(result) : 1 if success, 0 if busy, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
getState		int(error) : 0 if no error, 1 if error str(message) : optional message array(result) : [loading] [runningState] [paused] [duration] [position] [progress] [strPosition] [strDuration] [filename] [volume], -1 if error
getStrDuration		str(result) : h:i:s
stop		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
getProgress		int(result) : progress in pourcent, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
getDuration		int(result) : duration in secondes
togglePause		int(result) : 1 if paused, 0 if playing, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
seekBack30		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
setVolumeUp		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message

seekForward30		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
isRunning		int(result) : 1 if running, 0 if not running
getMediaInfo		str(result) : width x height, of media
getPauseState		int(result) : 1 if paused, 0 if playing
getStrPosition		str(result) : h:i:s
speedForward		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message

API i2c_lcd

Function	Params	Return
writeChar	str(char) : char to write to lcd	int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
writeString	str(string) : string to write to lcd	int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
cursorOff		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
blinkOff		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
home		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
blinkOn		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
backLightOn		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
setPosition	int(line) : optional, default is 0 int(position) : optional, default is 0	int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error

		str(message) : optional message
cursorOn		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
backLightOff		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
clear		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
write	str(message) : string to write to lcd with for new line	int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
start		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
command	str(data) : command to execute to lcd	int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message

API system

Function	Params	Return
getRpiLicence	str(name) : name can be "MPG2" or "WVC1"	int(error) : 0 if no error, 1 if error str(message) : optional message str(result) : "enabled" or "disabled", -1 if error
diskSpace	str(name) : optional mount name, default is "rootfs"	int(error) : 0 if no error, 1 if error str(message) : optional message array(result) : [int(used)] [int(free)] [int(pourcentUsed)] [int(pourcentFree)] if success, -1 if error
reloadApi		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
shutdown		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
shellExecute	str(command) : string command line to execute	int(error) : 0 if no error, 1 if error str(message) : optional message

		str(result) : <i>result of command, -1 if error</i>
restart		int(result) : <i>1 if success, -1 if error</i> int(error) : <i>0 if no error, 1 if error</i> str(message) : <i>optional message</i>
getRpiSerial		int(error) : <i>0 if no error, 1 if error</i> str(message) : <i>optional message</i> str(result) : <i>RaspBerry serial number; -1 if error</i>

API google speech

Function	Params	Return
isSpeaking		int(result) : <i>1 if success, -1 if error</i>
speak	str(lang) : <i>optional, else use default</i> str(text) : <i>string to speak</i> int(volume) : <i>optional, default is 100 pourcent</i>	int(result) : <i>1 if success, -1 if error</i> int(error) : <i>0 if no error, 1 if error</i> str(message) : <i>optional message</i>

API sound

Function	Params	Return
play	str(filepath) : <i>full filepath</i> int(volume) : <i>optional, default is 50 pourcent</i>	int(result) : <i>1 if success, 0 if already playing, -1 if error</i> int(error) : <i>0 if no error, 1 if error</i> str(message) : <i>optional message</i>
getVolume		int(error) : <i>0 if no error, 1 if error</i> str(message) : <i>optional message</i> array(result) : <i>[left] [right] [range] , -1 if error</i>
setMute	int(value) : <i>1 for disable, 0 for enable</i>	int(result) : <i>1 if success, -1 if error</i> int(error) : <i>0 if no error, 1 if error</i> str(message) : <i>optional message</i>
playIsRunning	str(filepath) : <i>optional, full filepath</i>	int(result) : <i>1 if filepath playing, 0 if not playing, if no filepath return number of file playing, -1 if error</i> int(error) : <i>0 if no error, 1 if error</i> str(message) : <i>optional message</i>
setOutput	int(output) : <i>1 analog output, 2 hdmi output, 0 autodetect</i>	int(result) : <i>1 if success, -1 if error</i> int(error) : <i>0 if no error, 1 if error</i> str(message) : <i>optional message</i>
setVolume	int(value) : <i>between 0 and 100</i>	int(result) : <i>1 if success, -1 if error</i> int(error) : <i>0 if no error, 1 if error</i> str(message) : <i>optional message</i>

API api

Function	Params	Return
getApi		int(count) : <i>number of result</i> array(result) : <i>list of api</i>
help	str(api) : <i>optional, filter api name, default list all help</i>	int(count) : <i>number of result</i>

	str(func) : optional, filter api function, default list all function	array(result) : list of help api
getApiEnabled		int(count) : number of result array(result) : list of enabled api
getApiDisabled		int(count) : number of result array(result) : list of disabled api

API plugins

Function	Params	Return
start	str(pluginName) : name of plugin to start	int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
schedulerStop		int(result) : 1 if success else 0 str(message) : optional message
schedulerStart		int(result) : 1 if success, 0 if already started str(message) : optional message
isRunning	str(pluginName) : name of plugin to test	int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
stopAll		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
stop	str(pluginName) : name of plugin to stop	int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
running		int(error) : 0 if no error, 1 if error str(message) : optional message array(result) : list of running plugin, -1 if error
schedulerIsRunning		int(result) : 1 if success else 0 str(message) : optional message

API webcam

Function	Params	Return
start		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
stop		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message

API mail

Function	Params	Return
getUnreadId	str(name) : optional mail account name, else read default account str(folder) : optional account folder, default is "INBOX"	int(error) : 0 if no error, 1 if error int(count) : number of unread mails array(result) : list of unread ids mails, -1 if error str(message) : optional message
getAccounts		int(error) : 0 if no error, 1 if error int(count) : accounts number array(result) : list of accounts, -1 if error str(message) : optional message
getUnreadMail	int(getContent) : optional, can be 0 or 1, default is 0 str(name) : optional mail account name, else read default account int(getSubject) : optional, can be 0 or 1, default is 1 str(folder) : optional account folder, default is "INBOX" int(getSingleId) : optional, single id of mail to read int(setAsRead) : optional, can be 0 or 1, default is 0	int(error) : 0 if no error, 1 if error int(count) : number of unread mails array(result) : array of results, -1 if error str(message) : optional message
getAccountFolders	str(name) : optional mail account name, else read default account	int(error) : 0 if no error, 1 if error int(count) : folders number array(result) : list of folders, -1 if error str(message) : optional message
countUnread	str(name) : optional mail account name, else read default account str(folder) : optional account folder, default is "INBOX"	int(result) : number of unread mails, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message

API socket_server

Function	Params	Return
start		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
stop		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
send	int(result) : optional str(message) : optional str(function) : optional str(name) : optional int(error) : optional, can be set to 0, 1 or -1	int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error str(message) : optional message
restart		int(result) : 1 if success, -1 if error int(error) : 0 if no error, 1 if error

		str(message): optional message
--	--	--

API i2c_servopro

Function	Params	Return
setSpeed	int(pin): value between 0 and 11 int(value): value between 0 and 250	int(result): 1 if success, -1 if error int(error): 0 if no error; 1 if error str(message): optional message
getAllPosition		int(error): 0 if no error; 1 if error str(message): optional message array(result): array of 12 values, -1 if error
setMotors	int(pin2): value between 0 and 11 int(pin1): value between 0 and 11 int(value1): value between 0 and 255	int(result): 1 if success, -1 if error int(error): 0 if no error; 1 if error str(message): optional message
isEnabled		int(result): 1 if enable, 0 if disable
setMode	int(value): optional, 0 for standard (90°), 1 for extended (180°), default is 1	int(result): 1 if success, -1 if error int(error): 0 if no error; 1 if error str(message): optional message
setPosition	int(pin): value between 0 and 11 int(value): value between 0 and 255 int(speed): optional, value between 0 and 250	int(result): 1 if success, -1 if error int(error): 0 if no error; 1 if error str(message): optional message
setEnabled	int(value): 0 = all off, 1 = all on	int(result): 1 if success, -1 if error int(error): 0 if no error; 1 if error str(message): optional message
getPosition	int(pin): value between 0 and 11	int(result): value of position, -1 if error int(error): 0 if no error; 1 if error str(message): optional message