JackBord®

Command Reference

JackBord Commands

Contents

Introduction

This is a reference of the various commands available on the JackBord.

Commands are used primarily on the command page but can also be used in programs on the PROG page.

General Commands

These are general commands for use with the JackBord.

bye

Put the JackBord to sleep as if you pressed the power

Purpose

Put the JackBord to sleep. Same as jbsleep command.

hi

Display a simple hello message, flashes the USER LEDs

Purpose

Lets you check that your properly connected to your JackBord.

vb

Display the status of the battery and power supply in general

Purpose

Displays status information for the battery including it charge state. Also shows the state of the power busses.

Syntax

vb

Example

```
*** Battery *** 3
Charge   : 93.0 %
Charger  : 1

** FAULTS ** 0 = OK  1 = FAULT
Bus 3a     : 0
Bus 3b     : 0
Bus 3c     : 0
Bus 5a     : 0
Battery    : 0
Charger    : 0
```

vc

View command system status

Purpose

Displays the status of the command interpreter.

Syntax

vc

Example

```
******** Command System Status ********
command                : vc
field no               : 1
Last cmd time          : 1 secs
Last rec cmd           : 0 secs
Poff thresh            : 0 secs
next_command           :
final_stack_command  : vc
in_turn_func_exits   : 0
final_stack_command_block :


MQTT  RX       : 3
      TX       : 6
CMD msg count  : 3
CI calls       : 3
Turns          : 0
cmd_debug_mode : 0
cmd_depth_gauge :1
```

vcs

View the command stack

Purpose

View the commands currently in the command stack.

Syntax

vcs

Example

```
> vcs
< ***** JackBord Command Stack [1] ********
GCSTK> cmd_stack[1] = [vcs]

< GCSTK> END cmd_stack[] ************
```

vjs

Send the status as a Json string

Purpose

Send status information as a Json string to the jstatus mqtt topic.

Syntax

vjs

Example

```
{"mplc":"0","sw_ver":"jackbord_ver29_28_June_
2020","sw_ver_no":"39.2","jbid":"jack.3","res
et":"0","suspect":"0","up":"3","jbip":"","lcm
d_ip":"","port":"3","wclients":"0","tree_mode
":"0","edit_user_prog_no":"1","boot_user_prog
_no":"0"}
```

vs

Display the status of the JackBord

Purpose

Displays status information for the JackBord including the up time and the running program etc.

Syntax

vs

I/O Commands

These commands allow you to control the various pins on the JackBord.

#~

Turn pin on or off

Purpose

Allows you to turn a given port A, C, D, E pin on or off.

Syntax

#~ state

Arguments

#

The port letter a, b, c, d

~

The port pin number 1 to 5

state

The desired state of the pin.
0 = OFF 1 = ON

Pushers No

Returns Nothing

Examples

a1 1	Turn port A pin 1 on
a1 0	Turn port A pin 1 off

Notes

This can be used from the command line and in programs.

Channels & Vars

Updates the corresponding channel.

#on

Turn all of port #'s pins on.

Purpose

This will turn all of the pins on the specified port on.

Syntax

#on

Arguments

#

The port letter a, c, d, e

Examples

aon	Turn all port A pins on
con	Turn all port C pin on

Notes

See the #off command to turn port pins off.

#off

Turn all of port #'s pins off.

Purpose

This will turn all of the pins on the specified port off.

Syntax

#off

Arguments

#

The port letter a, c, d, e

Examples

aoff	Turn all port A pins off
coff	Turn all port C pin off

Notes

See the #on command to turn port pins on

gav

Get Analog Voltage Reading

Purpose

gav gets a voltage reading from an analog input between 0 and 2.5V. You can also scale it to different ranges.

Syntax

gav source min max

Arguments

source

The pin no or channel where the voltage is.

min

The minimum scale value

max

The maximum scale value

Examples

gav b1

Gets the voltage on b1 between 0V and 2.5V.

gav b1 0 5.0

Gets the voltage on b1 and scales it between 0V and 5.0V.

get

Turns the pin into a logic input and gets a reading from it

Purpose

Set the pin as a logic input and get a fresh reading from it in real time.

Syntax

get #~

Arguments

#

The port letter a, b, c, d

~

The port pin number 1 to 5

Notes

Takes a direct digital reading from the hardware pin.

glr

Get a fresh set of readings and display them

Purpose

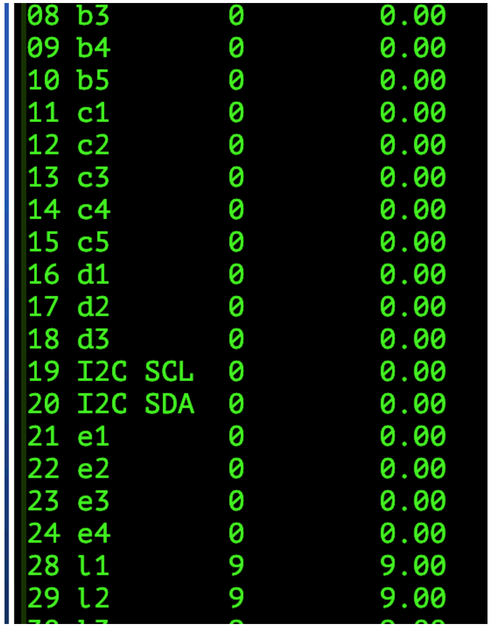
This will get fresh readings from any sensors etc and display them.

Syntax

glr

Examples

glr



```
08 b3        0       0.00
09 b4        0       0.00
10 b5        0       0.00
11 c1        0       0.00
12 c2        0       0.00
13 c3        0       0.00
14 c4        0       0.00
15 c5        0       0.00
16 d1        0       0.00
17 d2        0       0.00
18 d3        0       0.00
19 I2C SCL   0       0.00
20 I2C SDA   0       0.00
21 e1        0       0.00
22 e2        0       0.00
23 e3        0       0.00
24 e4        0       0.00
28 l1        9       9.00
29 l2        9       9.00
```

gqr

Get quick readings in one line form

Purpose

Gets a set of readings and puts them on one line.

Syntax

gqr

Examples

```
> gqr
<  a1 000 a2 000 a3 000 a4 000 a5 000
b1 000 b2 000 b3 000 b4 000 b5 000
c1 000 c2 000 c3 000 c4 000 c5 000 d1
000 d2 000 d3 000 d4 000 d5 000 e1
000 e2 000 e3 000 e4 000  000  000
000  000  000  000  000  000  000  000
000  000  000  000  000  000
```

grd

Get raw readings from the channels in one line form

Purpose

Gets a fresh set of raw readings and displays them on one line.

Syntax

grd

Examples

```
> grd
< 1 0 0.00|2 0 0.00|3 0 0.00|4 0 0.00|5 0
0.00|6 0 0.00|7 0 0.00|8 0 0.00|9 0
0.00|10 0 0.00|11 0 0.00|12 0 0.00|13 0
0.00|14 0 0.00|15 0 0.00|16 0 0.00|17 0
0.00|18 0 0.00|19 0 0.00|20 0 0.00|21 0
0.00|22 0 0.00|23 0 0.00|24 0 0.00|25 0
0.00|26 0 0.00|27 0 0.00|28 0 0.00|29 0
0.00|30 0 0.00|31 0 0.00|32 0 0.00|33 0
0.00|34 0 0.00|35 0 0.00|36 0 0.00|37 0
0.00|38 0 0.00|39 0 0.00|40 0 0.00|
```

Notes

The format of the output is:

|chan raw raw_float|

grrd

Get real raw readings from the channels in one line form

Purpose

Gets a fresh set of raw readings and displays them on one line.

Syntax

grrd

Examples

```
> grrd
<  a1 0 a2 0 a3 0 a4 0 a5 0 b1 0 b2 0 b3
0 b4 0 b5 0 c1 0 c2 0 c3 0 c4 0 c5 0 d1
0 d2 0 d3 1 d4 0 d5 0 e1 0 e2 0 e3 0 e4
0
```

Notes

The format of the output is:

pin value

gt

Get the values of the specified port.

Purpose

Returns a table of the current values of the port #. If the # port is left out it return the values of all ports.

Syntax

gt #   or gt

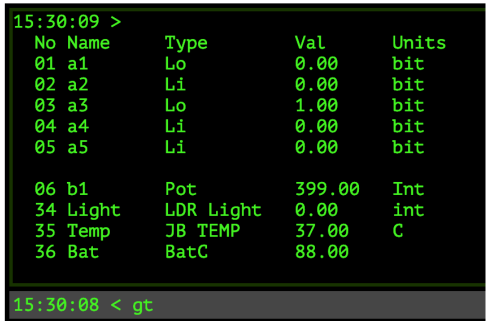
Arguments

#

The port letter a, b, c, d, e

Examples

gt



```
15:30:09 >
  No Name      Type        Val       Units
  01 a1        Lo          0.00      bit
  02 a2        Li          0.00      bit
  03 a3        Lo          1.00      bit
  04 a4        Li          0.00      bit
  05 a5        Li          0.00      bit

  06 b1        Pot         399.00    Int
  34 Light     LDR Light   0.00      int
  35 Temp      JB TEMP     37.00     C
  36 Bat       BatC        88.00

15:30:08 < gt
```

gt a	Get the port A values
gt 6	Get the value of channel 6

Notes

The type column shows the current type of the pin.

gvr

Get Voltage Reading

Purpose

gvr gets a voltage reading from an analog input on port B and converts into a value between **min** and **max**.

Syntax

gvr source min max dest

Arguments

source

The pin no or channel where the voltage is.

min

The minimum result value

max

The maximum result value

dest

The destination channel or variable where the voltage reading will be sent.

Returns

The scaled voltage reading as an integer or floating point value depending on the data type of the variable after the pusher.

Examples

gvr b1

gvr b1 0 100

gvr b2 0 100 a1

gvr b2 -50 50 160

jsga

Get readings from the channels and send as a Json data set via mqtt.

Purpose

Gets a fresh set of raw readings and puts them in a Json string and sends them via mqtt.

Syntax

jsga

Examples

jsga

```
{ "data":{"channels":[
{"chano":1,"name":"a1","type":0,"value":"0"}
,{"chano":2,"name":"a2","type":1,"value":"0"}
,{"chano":3,"name":"a3","type":0,"value":"0"}
,{"chano":4,"name":"a4","type":1,"value":"0"}
,{"chano":5,"name":"a5","type":1,"value":"0"}
,{"chano":11,"name":"c1","type":1,"value":"0"}
,{"chano":12,"name":"c2","type":1,"value":"0"}
,{"chano":13,"name":"c3","type":1,"value":"0"}
,{"chano":14,"name":"c4","type":1,"value":"0"}
,{"chano":15,"name":"c5","type":1,"value":"0"}
,{"chano":16,"name":"d1","type":1,"value":"0"}
,{"chano":17,"name":"d2","type":1,"value":"0"}
,{"chano":18,"name":"d3","type":1,"value":"0"}
,{"chano":19,"name":"d4","type":1,"value":"0"}
,{"chano":20,"name":"d5","type":1,"value":"0"}
,{"chano":21,"name":"e1","type":13,"value":"0"}
,{"chano":22,"name":"e2","type":13,"value":"0"}
,{"chano":23,"name":"e3","type":13,"value":"0"}
,{"chano":24,"name":"e4","type":13,"value":"0"}
]}}
```

pulse

Pulse Output

Purpose

Pulses the selected output on for the specified length of time.

Syntax

pulse #~ time

Arguments

#

The port letter a, b, c, d

~

The port pin number 1 to 5

time (milli secs)

The amount of time for which the pin will remain high.

Pushers No

Examples

pulse a1 100	Pulse a1 on for 100ms

sav

Set Analog Output Voltage

Purpose

Port pins B3, B4, E5 & E6 can be used to produce an analog voltage output.

Syntax

sav #~ value

Arguments

#

The port letter b or e

~

The port pin number 3,4,5 or 6

value

The desired output voltage 0 to 255 which is scaled between 0V and ~3V

Pushers No

Examples

sav b3 255

Notes

The amount of current provided on these pins when used as an analog output is low.

tg #~

Toggle the value of the selected pin.

Purpose

Toggles the value of the selected pin. If it's on it will go off and vice versa.

Syntax

tg #~

Arguments

#

The port letter a, b, c, d

~

The port pin number 1 to 5

Pushers Yes

Returns Nothing

Examples

tg a1	Toggle value of a1
tg d1	Toggle the value if the USER LED

tg d1 -> /user_led_state

updc

Manually do an update of all channels from hardware

Purpose

This calls the update_chan_values() function which updates the channels.

Syntax

updc

Notes

Testing only.

spm

Set a pins mode of operation

Purpose

Set the mode of operation of a pin.

Syntax

spm #~ mode

Arguments

#

The port letter a, b, c, d

~

The port pin number 1 to 5

mode

The mode of operation for the pin.

Mode	Description
0	Logic input no pull-up
1	Logic input with pull-up
2	Logic output
3	Analog input port b only.

Notes

spm a1 1
Set pin a1 as a logic input with a pull-up.

spm a4 2
Set pin a4 as a logic output.

Date & Time Commands

These are various date and time related commands.

dly

Delay for a set period of milli seconds

Purpose

Delays the execution of a program by the specified number of milli-seconds.

Syntax

dly delayms

Pushers No

Returns Nothing

Examples

dly 100	wait 100ms
dly 1000	wait 1sec

Notes

This command does not delay the command line or other functions when run in a program.

Standard Delays

The following is a table of standard delays built into JackBord.

Command	**Delay milli-secs**
d50	50
d100	100
d250	250
d500	500
d1000	1000
d1500	1500

Examples

d50	Wait 50ms
d500	Wait 0.5secs

System Timer Commands

rtimer1
rtimer2
rtimer 3

Reset system timer no 1 2 or 3

Purpose

Reset system timer no 1 back to 0.
This timer counts in milli-seconds from the time it was last reset and is available via the /timer1 system variable.

Syntax

rtimer1

Notes

The format is the same for the other 3 system timers.

eg rtimer2 and rtimer3

Smart LED Commands

These commands are for use with smart LED strips connected to the SLED pin (E1 on port E).

sledn

Set the number of smart LEDs connected to the sled pin on port pin E1.

Purpose

Tell the JackBord how many smart LEDs are connected to it on pin E1.

Syntax

sledn no

Arguments

no

The number of LEDs

Pushers No

Examples

sledn 10

Notes

Works with a WS2812B smart led strip.

LED Colours

These tables list the color numbers to use with the sled command. The second table provides the rgb values.

Color	Number
Off	0
Dim white	1
Bright white	2
Red	3
Orange red	4
Orange	5
Gold	6
Yellow	7
Green	8
Lime	9
Light green	10
Blue	11
Cyan	12
Navy	13
Magenta	14
Purple	15

RGB Colour Values

Color	Number	Red	Green	Blue
Off	0	0	0	0
Dim white	1	128	128	128
Bright white	2	255	255	255
Red	3	255	0	0
Orange red	4	255	69	0
Orange	5	255	165	0
Gold	6	255	215	0
Yellow	7	255	255	0
Green	8	0	128	0
Lime	9	0	255	0
Light green	10	144	238	144
Blue	11	0	0	255
Cyan	12	0	255	255
Navy	13	0	0	128
Magenta	14	255	0	255
Purple	15	128	0	128

sled

Smart LED control

Purpose

Control the color and brightness of an WS2812B Smart LED/strip

Syntax

sled led no color
sled led no R G B

Arguments

led no

The number of the LED to be controlled.
99 = all of them.

color

The color number from 0 to 15. See the table on the previous page.

R

The red value, 0 to 255

G

The green value, 0 to 255

B

The blue value, 0 to 255

Pushers No

Examples

sledn 10

sled 5 1

sled 5 255 255 0

Notes

Works with a WS2812B smart led strip.

sledb

Smart LED brightness control

Purpose

Control the brightness of an WS2812B Smart LED/strip

Syntax

sledb brightness

Arguments

brightness

The desired brightness of the LEDs between 0 and 100.

Pushers No

Examples

sledb 26

sledoff

Turn all of the smart LEDS off

Purpose

Turn all of the WS2812B Smart LEDs off.

Syntax

sledoff

Notes

Works with a WS2812B smart led strip.

PWM Commands
(Pulse Width Modulation)

These commands are for pulse width modulation.

lpwc

List the PWM system channel allocations

Purpose

List the PWM channel allocations on the esp32. The esp32 can handle a maximum of 6 pwm chans.

Syntax

lpwc

Examples

```
> lpwc
< PWM Chan      Chan    Name
1               1       a1
2               2       a2
3               16      d1
4               0
5               0
6               0
7               22      e2
8               24      e4
```

The last two pwm channels are used by the drive motors.

pwmi

List the PWM system settings info

Purpose

Display the settings being used by the PWM system.

Syntax

pwmi

Examples

```
******** PWM Info ********


PWM Freqs
Port          :50
Drive Motors  :1000
Pwm Bits      :10
Max PWM Duty Cycle    :1000
rc_servo_min_duty     :1700
rc_servo_max_duty     :8000
```

mfreq

Set PWM freq for the Drive Motors

Purpose

This sets the pwm frequency for the main drive motors.
NOTE: the updated freq gets picked up by the relevant functions.

Syntax

mfreq freq

Arguments

freq
The frequency to use for the pwm main drive system. 50 to 5000Hz.

pfreq

Set PWM freq for the I/O Ports

Purpose

This sets the pwm freq for the pwm command.

Syntax

pfreq freq

Arguments

freq
The frequency to use for the pwm system. 50 to 5000Hz.

pwm

Pulse width modulation on a pin

Purpose

pwm allows a pulse width modulated (PWM) signal to be applied to a pin.

Syntax

pwm #~ duty

Arguments

#~
The port letter a, b, c, d and pin number 1 to 5. eg a1 or c4

duty
The duty cycle from 0 to 100.

Pushers No

Returns Nothing

Examples

pwm a1 50
Set pin a1 at 50%

pwm d1 75
Set the USER LED at 75% brightness

pwm a3 /speed
Use the /speed variable to set the pwm duty on pin a3

A/D Converter Commands

These commands are for use with pins set to be analog inputs.

adcal

Set the A/D Converter gradient and offset values

Purpose

Use this to set a gradient and offset for the JackBord's A/D converter. This will be used on all analog readings.

Syntax

adcal grad offset

Arguments

grad
A floating point gradient value.

offset
A floating point offset value.

Examples

adcal 3.3 12

Notes

Don't use this unless you know what you are doing.

admax

Set the A/D Converter Max Voltage Scale Value

Purpose

If this is not 0 it is used to set the Max voltage that the A/D converter 4096 reading represents.

Syntax

admax value

Arguments

value

A floating point voltage value, 0 to 100

Notes

Don't use this unless you know what you are doing.

adwmax

Set the A/D converters ad_max_raw_value

Purpose

Sets the max digital reading we will get at full scale. This is usually
10 bits = 1024
12 bits = 4096
14 bits = 8092

Syntax

adwmax value

Arguments

value

A positive integer value, between 1000 and 8092.

Notes

Don't use this unless you know what you are doing.

1Wire Commands

These commands are for 1wire devices attached to the 1wire port on port E.

s1won

Enable the 1wire pin on port E.

Purpose

This turns on the JackBords 1wire system.

Syntax

s1won

s1woff

Disable the 1wire pin on port E.

Purpose

This turns off the JackBords 1wire system.

Syntax

s1woff

f1w

Find 1wire devices on the 1wire bus.

Purpose

This will scan for 1wire devices on the 1wire bus and add them to the JackBord.

Syntax

f1w

l1w

List One Wire devices attached to the JackBord

Purpose

This lists all of the Dallas One Wire devices attached to the JackBord and shows their readings.

Syntax

l1w

rst1w

Reset the 1wire system

Purpose

Resets the 1 wire sytem.

Syntax

rst1w

SPI Port Commands

Commands for the SPI ports.

spion

Enable the SPI system

Purpose

Turn on the SPI system.

Syntax

spion

spioff

Disable the SPI system

Purpose

Turn off the SPI system.

Syntax

spioff

spipins

Set the port pins to use for the SPI system

Purpose

Allows the pins to be used for the SPI system to be set.

Syntax

spipins SCK MISO MOSI CS

Arguments

SCK

Port pin to be used for the SCK pin.

MISO

Port pin to be used for the MISO pin.

MOSI

Port pin to be used for the MOSI pin.

CS

Port pin to be used for the CS pin.

Examples

spipins c2 c3 c4 c5

Setup port C pins 2 to 5 to be the SPI port pins.

Notes

The port B and E pins cannot be used for SPI.

spot

Set the Resistance value of the MCP4161 Digital Potentiometer

Purpose

Set the Resistance value of the MCP4161 Digital Potentiometer which is connected to the SPI port.

Syntax

spot value

Arguments

value

Digital pot value between 0 and 255.

Examples

spot 50

Notes

Connection to JackBord SPI Port on Port C.

Max MCP4161		JB SPI	PIN
CS	pin 1	CS	A5
SCK	pin 2	SCK	A2
SDI/SDO	pin 3	MOSI	A4

vspi

Display the status of the SPI system

Purpose

Displays the status and basic settings of the SPI port system.

Syntax

vspi

I2C Port Commands

These commands are for i1c pins on port D.

i2con

Enable the i2c system

Purpose

Turn on I2c System and update config.

Syntax

i2con

i2coff

Disable the i2c system

Purpose

Turn off I2c System and update config.

Syntax

i2coff

f2c

Find attached i2c devices

Purpose

This will scan the i2c port looking for attached i2c devices. When a new device is found it will be setup.

Syntax

f2c

Examples

```
> f2c
< Finding I2C Devices....

< Setup BME280 temp on next free chan [26]
< Setup BME280 RH on next free chan [27]
< Setup BME280 Pressure on next free chan [28]

Found 1 devices
```

Channels & Vars

Each of the devices found will be allocated the channels the require automatically. Their readings will appear in the gt command etc.

g2c

Get register value from an i2c device

Purpose

This will get the value of the specified register from the specified i2c device address.

Syntax

g2c addr reg bytes

Arguments

addr

The i2c devices address from the I2d command. 0 to 128

reg

The register number from 0 to 256

bytes

The number of bytes to expect back from the i2c device. 1 to 256.

Pushers Yes

Returns Nothing

Examples

```
> g2c 118 137 1
< g2c data 0[111] 1[0] final [111]
```

Get reg no 137, one bytes from i2c address 118.

Notes

This can be used from the command line and in programs.

I2c

List active i2c devices & sensors

Purpose

Lists all of the active i2c devices connected to the JacklBord's i2c port. This will list all of the devices associated with each i2c address.

Syntax

I2c

Examples

```
> I2c
< Chan  Name        Addr  Mode  Type     Units   Raw      Value
< 26   AirT         118   1      Air T    C       507744   16.19
< 27   RH           118   1      RH       %       57        57.11
< 28   Press        118   1      Press    mb       1113    1113.69
```

I2d

List active i2c devices only

Purpose

Lists all of the active i2c devices connected to the JacklBord's i2c port. It only lists whole devices, not sub devices.

Syntax

I2d

Examples

```
> I2d
<
I2c Devices
No   Chan  Name        Addr  Type
1    28    BME280      118   9
```

s2c

Send data to i2c device register

Purpose

Send data to the specified register on the specified i2c device. All values are integers.

Syntax

s2c addr reg value

Arguments

addr

The i2c devices address from the I2d command. 0 to 128

reg

The register number from 0 to 256

value

An integer value between 0 and 256.

Pushers Yes

Examples

```
> s2c 118 137 23 ->/result
```

Notes

This can be used from the command line and in programs.

v2c

Show the status of the i2c system

Purpose

Displays the status of the i2c system including the number of devices connected and the i2c pins.

Syntax

v2c

Examples

```
> v2c
********* I2C Status **********
Enabled                    1
Devices                    2
Clock Speed                100000
next_i2c_chan_no           3
GPIOs
SDA                        23
SCL                         22
```

Button Commands

These commands are for port pins used as buttons.

bp

Simulate a button press

Purpose

Simulates a button press on a given pin. This is used for the GCC version of Octagon or where a physical button is not available.

Syntax

bp #~

Arguments

#~

The port letter a, b, c, d and pin number 1 to 5. eg a1 or c4

Examples

```
bp a1
bp c4
```

Notes

This can be used from the command line and in programs.

Channels & Vars

Updates the corresponding channel.

br

Simulate a button release

Purpose

Simulates a button release on a given pin. This is used for the GCC version of Octagon or where a physical button is not available.

Syntax

br #~

Arguments

#~

The port letter a, b, c, d and pin number 1 to 5. eg a1 or c4

Examples

```
br a1
br c4
```

Notes

This can be used from the command line and in programs.

Channels & Vars

Updates the corresponding channel.

btd

Set the button debounce delay in milli seconds

Purpose

Sets the debounce delay for buttons.

Syntax

btd delay

Arguments

delay

The debounce delay in milli seconds. 50 to 250.

Examples

```
btd 50
```

Notes

Use with pins set as buttons.

btg

Get the state of a button

Purpose

Gets the state of the specified button

Syntax

btg #~

Arguments

#~

The port letter a, b, c, d and pin number 1 to 5. eg a1 or c4

Examples

```
> btg a1
< Btg button chan[1] pin [a1] State [1] 1 = pressed
```

btp

Set a button press event

Purpose

Set a command/s that should be run when the specified button is pressed.

Syntax

btp #~ command/s

Arguments

#~

The port letter a, b, c, d and port pin number 1 to 5

command/s

One or more commands that should be run when the button is pressed. Commands may be daisy chained.

Examples

```
btp a1 “Hi from button 1”
btp a2 d1 1|d500|d1 0
```

Notes

This can be used from the command line and in programs.

Drive Page Buttons:

The drive page buttons 1 to 9 are virtual buttons and are on port v ie v1 to v9.

eg btp v1 “Hi from drive button 1”

btr

Set a button release event

Purpose

Set a command/s that should be run when the specified button is released.

Syntax

btr #~ command/s

Arguments

#~

The port letter a, b, c, d and port pin number 1 to 5

command/s

One or more commands that should be run when the button is released. Commands may be daisy chained.

Examples

```
btr a1 “Bye from button 1”
btr a2 d1 1|d500|d1 0
```

Notes

This can be used from the command line and in programs.

Drive Page Buttons:

The drive page buttons 1 to 9 are virtual buttons and are on port v ie v1 to v9.

eg btr v1 “Bye from drive button 1”

lb

List buttons

Purpose

Display a list of the currently active buttons.

Examples

```
> lb
< Buttons
No   Chan  Name         Port
1    1    but a1       a1    CMD: "Button press a1"
                       Release "Button release a1"
```

lbc

List button commands

Purpose

Display a list of the commands required to reproduce the currently set buttons. Use this to backup button settings.

Examples

```
> lbc
< Button Commands
btp a1 "Button press a1"
btr a1 "Button release a1"
```

rstb

Reset all buttons

Purpose

reset all of the set buttons.

Motor and Driving Commands

These commands are for the built in motor drive pins on port E.

allstopon

STOP ALL MOVEMENT

Purpose

This stops all movement of the robot and puts it in a safe mode.

Syntax

allstopon

Notes

This command overrides ALL other commands that would normally be able to induce some kind of movement in the robots drive system.

The ONLY way to leave all stop on mode is to run the allstopoff command.

allstopoff

Leave all stop on mode.

Purpose

This causes the robot to leave all stop on mode.

Syntax

allstopoff

bms

Set the speed of both drive motors

Purpose

This sets the speed of both drive motors to the same value.

Syntax

bms value

Arguments

value

-100 to 100

Where:

-100	full reverse
0	stop
100	full forward

Pushers No

Returns Nothing

Examples

bms 0	Stop
bms 50	Forward half
bms -50	Back half

Channels & Vars

Channel **150** both motor speed
160 left motor speed
170 right motor speed

Notes

This is for robots with motor drives.

cbst

Cold start boost for the drive motors

Purpose

This is an extra duty cycle added to the motors speed when its starting from stand still.

Syntax

cbst value

Arguments

value

0 to 1000 This is a duty cycle.

Examples

cbst 100
Set to 100

Notes

This is for robots with motor drives.

dslimit

Disable Steering Limit Checking

Purpose

This allows the steering limit switch checking to be enabled and disabled on robots using a linear actuator with limit switches for steering.

Syntax

dslimit value

Arguments

value

0 = Don't disable
1 = Disable

Pushers No

Returns Nothing

Examples

dslimit 0
Don't disable steering limit switches

dslimit 1
Disable steering limit switches

Notes

This should only be used for testing.

lmd

Set Left Drive Motors Direction

Purpose

Set the direction of travel of the left drive motor.
0 = forward   1 = reverse

Syntax

lmd dir

Arguments

dir

0 = forward   1 = reverse

lmo

Set Left Drive Motors Offset

Purpose

This is added to the motors speed at a low level to compensate for differences in the motors.

Syntax

lmo offset

Arguments

offset

The duty cycle offset between 0 and 1000.

lms

Set the speed of the left drive motor

Purpose

This sets the speed of the left drive motor.

Syntax

lms value

Arguments

value

-100 to 100

Where:

-100	full reverse
0	stop
100	full forward

Pushers No

Returns Nothing

Examples

lms 0	Stop
lms 50	Forward half
lms -50	Back half

Channels & Vars

Channel **150** both motor speed
160 left motor speed

Notes

This is for robots with motor drives.

lmt

Left motor start threshold

Purpose

This is the min speed value at which the left motors wheels will turn.
Range 0 to 100.

Syntax

lmt value

Arguments

value

0 to 100

Examples

lmt 25

Channels & Vars

Channel **165**

Notes

This is for robots with motor drives.

rmd

Set Right Drive Motors Direction

Purpose

Set the direction of travel of the right drive motor.
0 = forward 1 = reverse

Syntax

rmd dir

Arguments

dir

0 = forward 1 = reverse

rmo

Set Right Drive Motors Offset

Purpose

This is added to the motors speed at a low level to compensate for differences in the motors.

Syntax

rmo offset

Arguments

offset

The duty cycle offset between 0 and 1000.

rms

Set the speed of the right drive motor

Purpose

This sets the speed of the right drive motor.

Syntax

rms value

Arguments

value

-100 to 100

Where:

-100	full reverse
0	stop
100	full forward

Examples

rms 0	Stop
rms 50	Forward half
rms -50	Back half

Channels & Vars

Channel **150** both motor speed
170 right motor speed

Notes

This is for robots with motor drives.

rmt

Right motor start threshold

Purpose

This is the min speed value at which the right motors wheels will turn.
Range 0 to 100.

Syntax

rmt value

Arguments

value

0 to 100

Examples

rmt 25

Channels & Vars

Channel **175**

Notes

This is for robots with motor drives.

rstdv

Reset the motor drive system

Purpose

Resets the drive system to the way it is when the JackBord is turned on.

Syntax

rstdv

Examples

rstdv

sbms

Set the speed of both drive motors to different values

Purpose

This sets the speed of both drive motors to the different values in the same command.

Syntax

sbms left right

Arguments

left

Speed of the left motor.
-100 to 100

Where:

-100	full reverse
0	stop
100	full forward

right

Speed of the right motor.
-100 to 100

Where:

-100	full reverse
0	stop
100	full forward

Pushers No

Returns Nothing

Examples

sbms 0 0	Stop
sbms 25 50	Left turn
sbms 50 25	Right turn
sbms 20 20	Forwards slow

Channels & Vars

Channel **150** both motor speed
160 left motor speed
170 right motor speed

Notes

This is for robots with motor drives.

sdmct

Set the JackBord drive motor controller type

Purpose

Set the type of motor being used on the robots drive system. Options are:

Type	Description
0	NO motor drive installed.
1	Normal MOSFET mode
2	DRV8833 H-bridge
3	L298N H-bridge
4	BTS7960B H-bridge
5	Mobility scooter drive using MCP4161 Digital Potentiometer
6	Single Front Wheel Drive on front. Connected to the L298 H-bridge Output 2.

Syntax

sdmct value

Arguments

value

Drive type type, see above.

Examples

sdmct 2
Set DRV8833 H-bridge type.

Notes

This is for robots with motor drives.

sdtur

Set default turn urgency

Purpose

Set the default turn urgency

Syntax

sdtur value

Arguments

value

1 to 50

Examples

sdtur 10
Set default turn urgency to 10

setstmax

Set the Maximum Safe Operating Speed for the robot

Purpose

This allows a safe maximum speed to be specified and it will be enforced automatically.

Syntax

setstmax speed

Arguments

speed

-100 to 100

Where:

-100	full reverse
0	stop
100	full forward

Notes

This is for robots with motor drives. The limit set here is automatically enforced by the drive system.

setstmin

Set the Minimum Safe Operating Speed for the robot

Purpose

This allows a safe minimum speed to be specified and it will be enforced automatically.

Syntax

setstmin speed

Arguments

speed

-100 to 100

Where:

-100	full reverse
0	stop
100	full forward

Notes

This is for robots with motor drives. The limit set here is automatically enforced by the drive system.

ssv

Set Speed Step Value

Purpose

Set the amount by which the speed of the drive motors will be changed by other commands such as turn.

Syntax

ssv value

Arguments

value

1 to 50

Pushers No

Returns Nothing

Examples

ssv 10
Set to 10

Notes

This is for robots with motor drives.

straight

Drive in a straight line

Purpose

Causes the robot to drive in a straight line.

Syntax

straight

stop

Stop both of the robots drive motors

Purpose

Use this to bring the robot to a halt.

Syntax

stop

Examples

stop
Halt the robot

Notes

This is for robots with motor drives.

strpw

Set the linear actuator pulse width in milli secs

Purpose

This sets the width of the pulses used to move linear actuators attached to the JackBord via an L298 H-bridge.

The shorter the pulse the slower and more precise the control is.

Syntax

strpw value

Arguments

value
Pulse width in milli-seconds between 1 and 1000ms.

Examples

strpw 200
Set the pulse width to 200ms

Notes

This is for robots with motor drives.

strtype

Set the steering type

Purpose

Set the type of steering being used on the robots drive system. Options are:

Type	Description
0	Normal differential drive
1	Rack and pinion steering
2	Single front steering wheel
3	L298 Linear Actuator ie Beach Robot. On L298 h-bridge motor no 1.

Syntax

stur value

Arguments

value
Steering type, see above.

Pushers No

Returns Nothing

Examples

strtype 0
Set differential steering

Notes

This is for robots with motor drives.

stur

Set turn urgency

Purpose

This is the increment used when turning. The higher the value the more aggressive turning will be.

Syntax

stur value

Arguments

value
Turn urgency 1 to 500. This is a duty cycle.

Pushers No

Returns Nothing

Examples

stur 200
Set the turn urgency to 200

Notes

This is for robots with motor drives.

tgdir

Toggle the direction of travel.

Purpose

This toggles the current direction of travel of the drive motors.

Syntax

tgdir

tsf

Set turn speed factor

Purpose

This is the increment used when turning. The higher the value the more aggressive turning will be.

Syntax

tsf value

Arguments

value
1 to 50

Pushers No

Returns Nothing

Examples

tsf 10
Set the turn speed factor to 10.

Notes

This is for robots with motor drives.

turn

Turn the robot left or right

Purpose

Provides a means of steering a robot using values in the range:

-100	Full left turn
0	Straight
100	Full right turn

The exact limits for left and right are set using the **minleft** and **minright** commands.

Syntax

turn value

Arguments

value

-100	Full left turn
0	Straight
100	Full right turn

Examples

turn 0
Go straight

turn -10
Soft left turn

turn 90
Hard right turn./

Channels & Vars

Channel **153**

Notes

This is for robots with motor drives.

vd

View drive system status

Purpose

View the status of the motor drive system.

Syntax

vd

Example

```
0> ******** Drive Status ********

Drive type [2] [DRV8833]
Throttle  L[00]-[00]R
Set Speed L[00]-[00]R
Actual    L[00] R[00]
NET       0
Set spd   0
Dir       0 L[0]-[0]R


SAFE Throttle Limts Min[-100]  Max[100]
LEFT> Act Spd       00  Set 00 dir 0
RIGHT> Act Spd      00  Set 00 dir 0


Steering type       0  Turn 0  prev 0 Trend 0
Steering Lock 0
Turn [0] prev [0]  Trend [0]  Steering Lock
[0] min_left_turn [-100] max_right_turn [100]


Settings ----->
PWM Chans Left    22  Right 24 for drv8833
Speed Step value  5  Offsets L 0 R 0
Start Thresholds  L 0  R 0  Cold Start boost
25 msecs

Default Turn Urgency 10  Turn urgency 5
turn_speed_factor 2


PWM Freqs Port    50 Drive Motors 1000  l pin
14  r pin 27  Pwm Bits 10


Debug mode 0
```

vds

View drive system settings

Purpose

View the settings of the motor drive system.

Syntax

vds

Example

```
0> ******** Drive Settings ********

Drive type [2] [DRV8833]
Throttle  L[00]-[00]R
Set Speed L[00]-[00]R
Actual    L[00]-[00]R
NET       0
Set spd   0
Dir       1 L[1]-[1]R


SAFE Throttle Limts Min[-100]  Max[100]
LEFT> Act Spd       00  Set 00 dir 1
RIGHT> Act Spd      00  Set 00 dir 1


Steering type       0  Turn 0  prev 0 Trend 0
Steering Lock 0


Turn [0] prev [0]  Trend [0]  Steering Lock
[0] min_left_turn [-100] max_right_turn [100]



Settings ----->
PWM Chans Left    22  Right 24 for drv8833
Speed Step value  5  Offsets L 0 R 0
Start Thresholds  L 0  R 0  Cold Start boost
25 msecs


Default Turn Urgency 50  Turn urgency 5
turn_speed_factor 2


PWM Freqs Port    50 Drive Motors 1000  Pwm
Bits 10


Debug mode 5
```

Servo Commands

These commands are for a linear actuator powered by an external h-bridge.

svp

Set servo position

Purpose

This sets the shaft position of an rc servo connected to the JackBord

Syntax

svp #~ pos

Arguments

#~
The pin to which the servo is connected to on ports A, C, D & E.

pos
The new shaft position in degrees from 0 to 180.

Examples

svp a1 0
svp a1 90
svp a1 180

svp /pin /pos
Set the servo on pin /pin to position /pos

svs

Sweep Servo from start to end

Purpose

Sweep the servo from 0 to the specified position and back. An optional delay between 1deg steps may also be added.

Syntax

svs #~ start end delay

Arguments

#~
The pin to which the servo is connected to on ports A, C, D & E.

start
The start position 0 -180deg.

end
The end position. 0 -180deg.

delay (optional)
This is an optional delay, in milli-seconds, between each 1deg step of the servo's shaft.
5 to 50,000 milli-seconds.

Examples

svs a1 0 90
svs a1 0 180 50

Network Commands

There are the various networking related commands.

apid

Set the Wifi access point SSID to use in World mode

Purpose

Sets the Wifi access point SSID that the JackBord will try to connect to when it boots into World mode.

Syntax

apid SSID

Arguments

SSID

The Wifi access point SSID. From 1 to 30 characters.

Examples

apid Tree
Set the SSID to Tree

apid Big Fat Cat
Set the SSID to Big Fat Cat

Notes

Check the current value with the **vn** command. Look for the Home SSID: entry.

appw

Set the Wifi access point password to use in World mode

Purpose

Sets the Wifi access point's password of the access point the JackBord will try to connect to in World mode.

Syntax

appw password

Arguments

password

The Wifi access point password. From 1 to 64 characters.

Examples

appw fish eat frogs
Set the access point password to fish eat frogs

Notes

You don't need quotes around the password.

fuserap

Force the use of the user set Wifi access point.

Purpose

If its set to 1 this means we want to force the JackBord to use the SSID set in the var even though the JBHub access point may also be available.

Syntax

fuserap value

Arguments

value

0 = Don't force the use of the users ap.
1 = Yes do force the use of the users ap

Examples

fuserap 0
fuserap 1

hget

Do an http request to the specified URL

Purpose

This will do an http GET request to the specified URL and store the results in the output variable, which must be of a text type.

Syntax

hget "URL" -> result

Arguments

"URL"

The http URL, in "" quotes to which the GET request is to be made.

result

This is a text variable that holds the result of the get request.

Examples

hget "www.google.com" -> /result

hspw

Set the treehouse mode Wifi access point password

Purpose

Sets the Wifi password to use when the JackBord is running in treehouse mode.

Syntax

hspw password

Arguments

password

The Wifi password. From 6 to 30 characters.

Examples

hspw fish eat frogs
Set the access point password to fish eat frogs

Notes

The Treehouse mode SSID will always be the JackBords factory ID, which is printed below port B.

jbtree

Set the JackBord to operate in Treehouse mode

Purpose

When the JackBord boots it will go into treehouse mode and remain there.

Syntax

jbtree

Notes

The JackBord will start a Wifi hotspot. Connect to it and use a web browser to visit 192.168.4.1

jbworld

Set the JackBord to operate in World mode

Purpose

When the JackBord boots it will go into world mode and remain there.

Syntax

jbworld

mqtusebk

Use the backup mqtt server

Purpose

Causes the JackBord to start using the backup mqtt server

Syntax

mqtusebk

phub

Ping the mqtt hub

Purpose

Sends a ping to the mqtt server Zak.

ping

Ping an ipv4 address

Purpose

Does an Internet ping to the ip address to see if it is up.

Syntax

ping ip

Arguments

ip

The ipv4 ip address to ping.

Returns
ping time msecs

Pushers
Yes

Examples

ping 8.8.8.8

Reply:

Ping 8.8.8.8 ave: 64 msecs

rstnw

Reset the network system

Purpose

Resets the network system to the way it is when the JackBord is turned on.

Syntax

rstnw

Examples

rstnw

src

Send command to a remote JackBord

Purpose

This command allows you to send a command to a remote JackBord as if you were running the command on it directly.

The jallow command must be run on the remote JackBord first to give permission for the commands to be sent.

Syntax

src jbid command/s

Arguments

jbid

The ID of the remote JackBord to whom we wish to send the command.

command/s

One or more commands to be sent. Commands can be daisy chained,

Examples

src jack.2 tg d1
src jack.2 print “This is good”|d1 1

vn

View network status

Purpose

Displays the status of the network system.

Syntax

vn

Example

```
***** World Mode Network Status *****


Connected to  : Tree
Home SSID     : Tree
Signal        : -18 dBm (-30 Amazing  -67 Very Good  -70 OK  -80 Bad
-90 Unusable)
JB ip         : 192.168.4.33


Wifi Mac      : 34:94:54:AF:16:40
Unique ID     : 349454AF1640
Short ID      : 1640
```

Mqtt Commands

These are for the MQTT system.

vq

Display the MQTT system status

Purpose

Show the MQTT systems current status and settings.

Syntax

vq

Examples

```
*** MQTT Status ***
MQTT On      : 1
Mqtt id      : jack.3
QOS          : 1
Up time      : 12
Mqtt Server  : 192.168.4.169
BK           : mqttb.jackbord.org
USE BK       : 0
Re-Connects  : 0
Failed cons  : 0
Send Errors  : 0
MQTT RX      : 2
TX           : 5
TX Rate      : 0
CMD msg count  : 2
CI calls       : 2
Debug mode     : 0
```

Octagon Commands

These commands are for use with the PROG page when your writing programs.

clearprog

Reset the selected program.

Purpose

Set the selected program to the default program.

Syntax

clearprog no

Arguments

no

The number, 1 to 10, of the program to clear.

99 will clear ALL 10 programs.

Notes

This sets the selected program to the default program code.

exitnow

Exit the currently running program now

Purpose

Unlike the exitprog command this will NOT complete the current cycle of the prog_loop task. Instead it will exit immediately and run the prog_stop task.

Syntax

exitnow

Examples

```
prog_loop =

    exitnow
    “This will NOT be run”

prog_loop.
```

exitprog

Exit the currently running program

Purpose

Completes the prog_loop task and then exits the prog_loop task and runs the prog_stop task, ending the program.

Syntax

exitprog

Examples

```
prog_loop =

    exitprog
    “This will still be run”

prog_loop.
```

hardstop

Stop the currently running program at the hardstop, don’t run prog_stop task

Purpose

This command will stop execution at the hardstop line and NOT run the prog_stop task.

Syntax

hardstop

Examples

```
prog_loop =

    hardstop
    “This will NOT be run”

prog_loop.
```

lrc

List run code

Purpose

Lists the code of the program in the POGE page on the command line.

Syntax

lrc

Examples

```
lrc
```

ltask

List the tasks in the current program

Purpose

Lists the tasks in the current program.

Syntax

ltask

Examples

```
ltask
```

patdog

Do nothing

Purpose

Harmlessly waste some time in a program.

patcat

Do nothing

Purpose

Harmlessly waste some time in a program.

pauseprog

Pause the currently running user program

Purpose

Allow the running program to be paused so that single stepping can be done.

Syntax

pauseprog

Notes

Use the single step button on the PROG page to step through program lines one by one.

To resume the program either click on the resume icon or run the resumeprog command.

pclear

Clear the Print tab on the PROG page.

Purpose

This command deletes any text on the Print tab on the PROG page.

Syntax

pclear

Example

```
pclear
```

pticks

Octagon main program loop run step value

Purpose

Changes the speed at which programs run on the PROG page. Smaller values are faster.

Syntax

pticks value

Arguments

value

1 to 50,000

Examples

```
pticks 100
Set to 100
```

quit

Exit the Virtual JackBord Octagon program

Purpose

Quit the GCC version of Octagon,

resumeprog

Resume a paused program

Purpose

This will resume execution of a paused program.

Syntax

resumeprog

Examples

```
resumeprog
Resume the paused program.
```

return

Return from the current user task

Purpose

Allows an exit from a user task before we reach the termination of the task.

Syntax

return

Examples

```
get_fish =
        "count /count"
        inc /count
        return
        "Will never get here"
        any /count > 10 -> skipout
get_fish.
```

runu

Run the specified user program

Purpose

Allow a user program to be run from the command line.

Syntax

runu prog no

Arguments

prog no

The program number from 1 to 10 to run.

Examples

```
runu 1
Run program no 1.
```

setboot

Set program to run at boot.

Purpose

Sets the number of the user program that will be automatically run when the JackBord starts up.

Syntax

setboot prog no

Arguments

prog no

The program number to run at boot from 1 to 10.

Examples

```
setboot 1
Set program 1 to run at boot.
```

Notes

Use prog no 0 to disable any program from running at boot.

singlestep

Run the next instruction of a paused program

Purpose

While a program is paused this command will run one instruction each time it is called. use this for debugging.

Syntax

singlestep

skipout

Exit from the current repeat statement

Purpose

Exit a repeat statement block before the set number of repeats has been completed.

Syntax

skipout

Examples

```
repeat 20 ->

        "count /count"
        inc /count
        any /count > 10 -> skipout
repeat.
```

stopu

Stop the currently running user program

Purpose

Allow the currently running user program to be stopped from the command line.

vp

View program status

Purpose

Displays the status of the current program.

Syntax

vp

Examples

```
> vp
2> T[0] PC[0] ******** User Program Status ********
Edit: 1
Run:  1
Boot: 0

State: Exe: 0  Pause: 0  Exit: 0  Lines: 0  Silent: 0 in_upl: 0
prog ticks: 5


Current line[0] Current task[0] Exit line[0]
Memory [3448] Free Heap [37660]  Min free heap size
[17280] Vars [1]
```

Variable Commands

These are commands for use with variables on the JackBord.

d/

Declare a new variable

Purpose

Declares a new variable and sets its initial value

Syntax

d/name value

Arguments

name

The new variables name. Maximum of 32 chars, all one word.

value

The default value of the new variable. This determines the variables data type.

d/sheep 0	Integer variable
d/height 0.0	Floating point variable
d/town “”	Text variable

Pushers No

Returns Nothing

Examples

d/count 1
d/first_name “Jack”
d/age 10.6

dc/

Declare a new control variable

Purpose

Declares a new control variable and sets its initial value. The value of the control variable will automatically update the value of the target channel.

Syntax

d/name chan value

Arguments

name

The new variables name. Maximum of 32 chars, all one word.

chan

The channel number the control variable will be linked to.

value

The default value of the new variable. This determines the variables data type.

d/sheep 0	Integer variable
d/height 0.0	Floating point variable
d/town “”	Text variable

Examples

dc/left_motor_speed 160 0
dc/right_motor_speed 170 0

lv

List Variables

Purpose

Lists all of the variables.

Syntax

lv

rstv

Reset Variables

Purpose

This will reset all variables currently in use.

Syntax

rstv

Address Book Commands

These commands are for the address book and allow you to make contacts for other JackBords you use.

cabe

Clear all address book entries

Purpose

Removes all entries in the address book.

Syntax

cabe

labe

List address book entries

Purpose

Lists the entries in the address book.

Syntax

labe

rabe

Remove address book entry

Purpose

Remove an entry from the address book. Use the entry no from the labe command to remove.

Syntax

rabe entry no

Arguments

entry no

The number of the address book entry to remove from the labe command.

Examples

rabe 3
Remove entry no 3.

Notes

Use prog no 0 to disable any program from running at boot.

sabe

Set address book entry

Purpose

Add a new entry to the address book. This includes the entry name and the ID of the JackBord the entry is for.

Syntax

sabe name JackBord ID

Arguments

name

The name for the new address book entry. Max 30 chars, all one word and no punctuation except the _ char.

JackBord ID

The ID of the remote JackBord the entry is for.

Examples

sabe controller dog.cat
Add a new entry called controller for the JackBord dog.cat

Notes

Use prog no 0 to disable any program from running at boot.

Text Commands

These are commands for use with text and include ways of printing.

jp

Print some text in the command line

Purpose

Print the text in the "" quotes on the command line.

Syntax

jp "text"

Arguments

"text"

The text to be printed.

Examples

jp "Hi from Octagon"
jp "Pi is /pi"

print

Print some text

Purpose

Print the text in the "" quotes in the PROG page Print window.

Syntax

print "text"

Arguments

"text"

The text to be printed.

Examples

print "Hi from Octagon"
print "Pi is /pi"

""

Print some text. Same as the print command.

Purpose

Print the text in the "" quotes in the PROG page Print window.
Don't need the print command.

Syntax

"text"

Syntax

"text"

Arguments

"text"

The text to be printed.

Examples

"Hi from Octagon"
"Pi is /pi"

Data Logging Commands

These are commands for data logging on the JackBord.

lgon

Turn data logging on

Purpose

Start the logging of data.

Syntax

lgon

lgoff

Turn data logging off

Purpose

Stop the logging of data.

Syntax

lgoff

slog

Set the logging interval

Purpose

Set the data logging interval in seconds between sample collections.

Syntax

slog interval

Arguments

interval

The interval between samples in seconds. 10 to 10,000 secs.

Examples

slog 60
Set the logging interval to 60 secs

System Commands

Various system commands for the JackBord.

doupnb

Update the JackBord to the latest firmware version on the next boot

Purpose

This updates to the latest firmware when the JackBord next boots.

Syntax

doupnb

Examples

doupnb

Notes

ONLY do this when the battery is at least half charged and you have a good Internet connection. DO NOT turn off the JackBord during update.

dsd

Display various types of system data

Purpose

This will display system data based upon the specified mode.

Syntax

dsd mode

Arguments

mode

0 = List options
1 = JackBord System Info
2 = raw A/D Data.
3 = Network Info
4 = Button info
5 = Pwm settings

Examples

```
> dsd 1
0> JackBord System Info -->
SW            jackbord_ver29_25_June_2020
SW ver no     39.2
HW            b2
JackBord HW   DOIT ESP32 DEVKIT V1
ESP chip type 16
Bat V     0.00
I drain   0.00 mA

Mpl Cycles
100   78
1K    378
5K    2378
10K   7378
50K   17378
Mpl       67378

MPL Runs msecs 4> 0 0 1 1  0 0 1 0 |
sw debug 0 jpr delay 0  I2c Enabled 1
Core 0 count 45127
Core 1 count 67378
```

jboff

Power off the JackBord

Purpose

Powers off the JackBord

Syntax

jboff

pofft

Set the auto power off time in minutes

Purpose

Sets the time in minutes after which the JackBord will power off in the event it does not receive any commands from the user.

Syntax

pofft value

Arguments

value

Time in minutes after which the JackBord will power down if no user commands are received. 1 to 1440 mins.

Examples

pofft 30
Auto turn off after 30 mins

reboot

Reboot the JackBord

Purpose

Reboots the JackBord

Syntax

reboot

resetjb

Reset the JackBord's various settings back to their defaults

Purpose

Resets many of the JackBords settings back to factory defaults.

Syntax

resetjb

Notes

ONLY do this if you know what you are doing.

savecfg

Save the JackBords configuration to flash memory

Purpose

Save the JackBords settings.

Syntax

savecfg

serspd1

Set the speed in bps of serial port 1

Purpose

Sets the speed pf serial port no 1.
Valid speeds include:

2400, 4800, 9600, 19200, 38400, 57600 and 115200

Syntax

serspd1 speed

Arguments

speed

The serial speed in bits per second.
Valid ranges see above.

Examples

serspd1 57600

up

Get the uptime in secs

Purpose

Display the time in seconds since the JackBord powered on.

Syntax

up

Examples

```
> up
< Up time 2351 secs
```

upbrain

Update the JackBord's brain to the latest firmware version

Purpose

This updates the brain to the latest firmware.

Syntax

upbrain

Notes

This can take a few minutes and you must NOT turn off the JackBord while the update process is underway.

upjb

Update the JackBord to the latest firmware version

Purpose

This updates to the latest firmware.

Syntax

upjb

Notes

ONLY do this when the battery is at least half charged and you have a good Internet connection. DO NOT turn off the JackBord during update.

Channel Commands

These commands are for use with channels.

chon

Enable the specified channel number.

Purpose

This will make the specified channel number active so it will appear in the gt command etc.

Syntax

chon channel

Arguments

channel

The channel number to enable.

Example

chon 1
Enable channel no 1.

choff

Disable the specified channel number.

Purpose

This will make the specified channel number in-active so it will not appear in the gt command etc.

Syntax

choff channel

Arguments

channel

The channel number to disable.

Example

choff 1
Disable channel no 1.

chup

Set the interval in milli-seconds between updates of the channels from hardware

Purpose

Controls the rate at which the channel values are updated from the associated hardware.

Syntax

chup delay

Arguments

delay

The delay in milli-seconds between updates. 25 to 60,000 milli-seconds.

Example

chup 100
Set update interval to 10ms.

delsync

Delete a sync entry from the sync table

Purpose

Allows you to delete a sync entry from the sync table thus disabling the sync.

Syntax

delaync no

Arguments

no

The entry no in the sync table to delete.

Example

delsync 1
Delete sync no 1

gc

Get the value of a channel

Purpose

Get the value of the specified channel.

Syntax

gc channel

Arguments

channel

The channel number to get.

Examples

```
> gc 1
< 01 0

> gc 240
< 240 = 68864 68864
```

gcs

Get channel settings

Purpose

Gets the current channel settings in various formats based upon the selected mode.

Syntax

gcs mode

Arguments

mode

<ode	Description
1	Send short short version
2	Send lots of detail
3	Send second half of detailed ver.
4	Send full schan command

Examples

gcs 1

gsc

Get a list of set channels, ie those that are not 0

Purpose

Display a list of all channels that are not 0.

Syntax

gsc

jallow

Allow another JackBord to control this one

Purpose

This command allows another JackBord to send it commands and to also sync channels and variables with it.
Remove this permission with the jremove command.

Syntax

jallow jbid

Arguments

jbid

The id of the remote JackBord that we wish to allow to control us.

Examples

jallow cat.dog
jallow mr.blamo

Notes

Use the jremove command to end the permissions.

jremove

Remove the ability of another JackBord to control this one

Purpose

This command reverses the effects of the jallow command.

Syntax

jremove jbid

Arguments

jbid

The id of the remote JackBord that we wish to no longer allow to control us.

Examples

jremove cat.dog
jremove mr.blamo

ls

List active syncs

Purpose

Displays a list of all active synchronizations.

Syntax

ls

rstc

Reset the channels

Purpose

Resets the channels to the way it is when the JackBord is turned on.

Syntax

rstc

Examples

rstc

rstsyncs

Delete all sync entries from the sync table

Purpose

Allows you to delete a sync entry from the sync table thus disabling the sync.

Syntax

rstsyncs

sc

Set the value of a channel

Purpose

Set the value of the specified channel.

Syntax

sc channel value

Arguments

channel

The channel number to get.

value

The new numeric value for the channel. Must be an integer or a floating point value.

Examples

sc 150 50
Set channel 150 to 50

schan

Setup a channel

Purpose

Add a new entry to the address book. This includes the entry name and the ID of the JackBord the entry is for.

Syntax

```
schan  chan
       name
       gpio_pin
       pin_name
       mode
       type
       units
       grad
       off
       min
       max
       vmin
       vmax
       cal1
       cal2
       cal3
       cal4
```

Arguments

chan	The channel no
name	User name for the chan.
pin_name	The pin name ie a1, b4,d3,e1 etc.
gpio_pin	GPIO pin on the esp32.
mode	0 = not in use 1 = in use
type	The type of pin:

Type	**Type**	**Description**
0	Logic I/O IN	Normal logic I/O pin. INPUT
1	Logic I/O OUT	Normal logic I/O pin. OUTPUT
2	Servo Pin	This pin has a servo on it
3	PWM pin	PWM drive pin
4	A/D Input	A/D Converter input
5	D/A Output	D/A Converter output
6	Switch input	This pin is a stnadard switch input.
7	Touch Switch input	The pin is a touch input on port B.
8	Counter	The pin is a counter inout.
9	Drive MOTORS	The pis is used for one of the drive motors.

units	The unit of measurement ie binary, TempC, RH%
grad	Gradient used to calc the final value for the chan
offset	Offset used to calc the final value for the chan
min	Current minimum reading from the channel
max	Current maximum reading from the channel
vmin	The minimum VALID reading for this channel
vmax	The maximum VALID reading for this channel
cal1	The first calibration value for this channel
cal2	The second calibration value for this channel
cal3	The third calibration value for this channel
cal4	The fourth calibration value for this channel

Notes

The grad, offset and calibration fields are all floating point values.

setuct

Set channel value change threshold to trigger an update

Purpose

The value of the channel must have changed by this or more in order for an mqtt update to be sent.

Syntax

setuct chan threshold

Arguments

chan

The channel number.

threshold

A floating point value by which the channel must have changed.

Examples

setuct 6 10
Chan 6 must change by 10 or more for an update to be sent.

sync

Synchronize channel or variable

Purpose

This command allows you to sync a local channel or variable to another local channel or variable. Or to sync the local value with the corresponding one on a remote JackBord.

Syntax

sync source dest jbid

Arguments

source

The source channel or variable that will be used to update the destination.

dest

The destination channel or variable that will be updated by the source.

jbid

The optional remote JackBord id. If this is set the source will be used to update the destination on the remote JackBord.

NOTE the jallow command must be run on the remote JackBord first.

Examples

sync b1 a1
Sync the value of b1 to a1

sync b1 a1 jack.2
Sync the value of b1 to pin a1 on the remote JackBord jack.2

Maths Commands

These are maths commands.

abs

Get the absolute value of a number

Purpose

The abs command finds absolute value of a number. Makes negative numbers positive.

Syntax

abs value -> /result

Arguments

value

A number

/result

The numeric variable that will hold the result.

Pushers Yes

Returns The numeric result of the operation.

Examples

```
abs -4 -> /total       (4)
abs /sin_value -> /sin_value
abs /negative_number -> /positive_number
```

add

Find the sum of two or more numbers

Purpose

This command will accept upto 20 numbers or variables and find their total. It will then store the result in the variable after the pusher.

If run on the command line it will display the result directly.

Syntax

add nn nn ... -> /result

Arguments

nn

Upto 20 numbers or numeric variables

/result

The numeric variable that will hold the result.

Pushers Yes

Returns The numeric result of the operation.

Examples

```
add 3 45 -> /total
add /total 1 -> /total
add 1 2 3 4 5 6 7 -> /total
add /temp 0.25 -> /temp
```

avg

Find the average value of the provided numbers

Purpose

Returns the average of the provided numbers.

Syntax

avg nn nn ... -> /result

Arguments

nn

Upto 20 numbers or numeric variables

/result

The numeric variable that will hold the result.

Pushers Yes

Examples

```
avg 9 1 34 2 -> /result
avg /a /b /c /d -> /result
avg 3.14156 2.9 4.2 1.2 -0.5-> /result
```

div

Divide the provided numbers.

Purpose

The div command divides the provided numbers moving from left to right.

Syntax

div nn nn ... -> /result

Arguments

nn

Upto 20 numbers or numeric variables

/result

The numeric variable that will hold the result.

Pushers Yes

Returns The numeric result of the operation.

Examples

```
div 5 10 -> /result
div /total 1 -> /result
div 1 2 3 4 5 6 7 -> /result
```

hash

Generate a unique integer number for the provided text

Purpose

The hash command generates a unique integer number for the provided text input.

Syntax

hash “text” -> /result

Arguments

text

A text string of upto 30 chars, must be in “” quotes.

/result

The numeric variable that will hold the result.

Pushers Yes

Returns The numeric result of the operation.

Examples

```
>  hash "jack and jill"
< Hash of "jack and jill" is 5193373875650668696

>  hash "jack and Jill"
< Hash of "jack and Jill" is 5193373875649518712
```

dec

Decrement a value

Purpose

This command will decrement the value fo the specified port or variable by 1. Or if an offset is provided by that offset.

Syntax

dec target offset

Arguments

target

This is the desired port pin ie a1, c4 etc. A channel number or a variable.

offset

The amount by which the value of the target is to be decremented. If this is omitted a value of 1 will be used by default.

Pushers Yes

Examples

dec a1
Decrement the value of port a1 by 1.

dec a1 5
Decrement the value of port a1 by 5.

dec /count
Decrement the value of the /count variable by 1

dec /count 100
Decrement the value of the /count variable by 100

dec /beans -> /bean_count

frac

Gives the fractional part of a floating point number.

Purpose

Calc the fraction part after the decimal point of a floating point number.

Syntax

frac value -> /result

Arguments

value

A number

/result

The numeric variable that will hold the result.

Examples

frac 123.456   gives 0.456

frac 1.123 gives 0.123

inc

Increment a value

Purpose

This command will increment the value fo the specified port or variable by 1. Or if an offset is provided by that offset.

Syntax

inc target offset

Arguments

target

This is the desired port pin ie a1, c4 etc. A channel number or a variable.

offset

The amount by which the value of the target is to be incremented. If this is omitted a value of 1 will be used by default.

Pushers Yes

Examples

inc a1
Increment the value of port a1 by 1.

inc a1 5
Increment the value of port a1 by 5.

inc /count
Increment the value of the /count variable by 1

inc /count 100
Increment the value of the /count variable by 100

inc /beans -> /bean_count

log

Calculates the base 10 logarithm of a number

Purpose

Calculate the base 10 log of a number.

Syntax

log value -> /result

Arguments

value

A number

/result

The numeric variable that will hold the result.

Pushers Yes

Returns The numeric result of the operation.

Examples

log 5 -> /result
log /total -> /result
log 0.5 -> /result
log 5   gives 0.698970

logn

Calculates the natural logarithm of a number

Purpose

Calculate the natural log of a number.

Syntax

log value -> /result

Arguments

value

A number number

/result

The numeric variable that will hold the result.

Pushers Yes

Returns The numeric result of the operation.

Examples

log 5 -> /result
log /total -> /result
log 0.5 -> /result
logn 10 gives 2.302585

max

Find the maximum value of the provided numbers

Purpose

Returns the highest of the provided numbers.

Syntax

max nn nn ... -> /result

Arguments

nn

Upto 20 numbers or numeric variables

/result

The numeric variable that will hold the result.

Pushers Yes

Returns The numeric result of the operation.

Examples

```
max 9 1 34 2 -> /result
max /a /b /c /d -> /result
max 3.14156 2.9 4.2 1.2 -0.5-> /result
```

min

Find the minimum value of the provided numbers

Purpose

Returns the lowest of the provided numbers.

Syntax

min nn nn ... -> /result

Arguments

nn

Upto 20 numbers or numeric variables

/result

The numeric variable that will hold the result.

Pushers Yes

Returns The numeric result of the operation.

Examples

```
min 9 1 34 2 -> /result
min /a /b /c /d -> /result
min 3.14156 2.9 4.2 1.2 -0.5-> /result
```

mul

Multiply the provided numbers.

Purpose

The mul command will multiply the provided numbers moving from left to right and store the product in the /result variable.

Syntax

mul nn nn ... -> /result

Arguments

nn

Upto 20 numbers or numeric variables

/result

The numeric variable that will hold the result.

Pushers Yes

Returns The numeric result of the operation.

Examples

```
mul 5 10 -> /result
mul /total 1 -> /result
mul 1 2 3 4 5 6 7 -> /result
mul 1 2 3 4 5 6 7   gives 5040
```

pow

Raise the base to a power

Purpose

The pow command will raise a base to a power.

Syntax

pow base power -> /result

Arguments

base

The base value that we will raise to the power

power

The power to which we will raise the base, can be neg or positive but not 0

/result

The numeric variable that will hold the result.

Pushers Yes

Returns The numeric result of the operation.

Examples

```
pow 5 10 -> /result
pow /total 0.4 -> /result
pow 2 /power -> /result
```

rand

Generate a random number between min and max

Purpose

Use the rand command to generate a random number between min and max

Syntax

rand min max -> /result

Arguments

min

The minimum desired random number

max

The maximum desired random number

/result

The numeric variable that will hold the result.

Pushers Yes

Returns The numeric result of the operation.

Examples

```
rand 1 10 -> /result
rand /min /max -> /result
```

round

Round a floating point number

Purpose

The round command takes a floating point number and rounds it to the specified number of decimal points.

Syntax

round value decimals -> /result

Arguments

value

A floating point number

decimals (optional)

The number of decimal places to round the number to, between 0 and 5.

This is an optional field. If omitted 2dp will be used.

/result

The numeric variable that will hold the result.

Pushers Yes

Returns The numeric result of the operation.

Examples

```
round 3.141 1    gives 3.1
round  3.9  0    gives 4
round /pi 3      gives 3.141
```

sqrt

Find the square root of a positive number

Purpose

Use the sqrt command to find the square rot of a number

Syntax

sqrt nn -> /result

Arguments

nn

A positive number above 0

/result

The numeric variable that will hold the result.

Pushers Yes

Returns The numeric result of the operation.

Examples

```
sqrt 9 -> /result
sqrt /total -> /result
sqrt 3.14156 -> /result
```

sub

Subtract the provided numbers.

Purpose

The sub command will work through the provided numbers from left to right, subtracting as it goes.

Syntax

sub nn nn ... -> /result

Arguments

nn

Upto 20 numbers or numeric variables

/result

The numeric variable that will hold the result.

Pushers Yes

Returns The numeric result of the operation.

Examples

```
sub 5 10 -> /result
sub /total 1 -> /result
sub 1 2 3 4 5 6 7 -> /result
```

tint

Return the integer part of a floating point number

Purpose

The tint command takes a floating point number and returns the integer part of the number.

Syntax

tint value -> /result

Arguments

value

A floating point number

/result

The numeric variable that will hold the result.

Pushers Yes

Returns The numeric result of the operation.

Examples

```
tint 3.141    gives 3
tint 3.9      gives 3
tint /pi      gives 3
```

Trigonometry

Trigonometry commands.

cos

Calculates the cosine of an angle in radians

Purpose

Calculates the cosine of the provided angle in radians.

Syntax

cos angle -> /result

Arguments

angle

Angle in radians.

/result

The numeric variable that will hold the result.

Pushers Yes

Returns The numeric result of the operation.

Examples

cos 45 -> /result (0.525322)

```
cos /pi -> /result
cos 45 -> /result
```

sin

Calculates the sine of an angle in radians

Purpose

Calculates the sine of the provided angle in radians.

Syntax

sin angle -> /result

Arguments

angle

Angle in radians.

/result

The numeric variable that will hold the result.

Returns The numeric result of the operation.

Pushers Yes

Examples

sin 45 -> /result (0.8509)

```
sin /pi -> /result
sin 45 -> /result
```

tan

Calculates the tan of an angle in radians

Purpose

Calculates the tan of the provided angle in radians.

Syntax

tan angle -> /result

Arguments

angle

Angle in radians.

/result

The numeric variable that will hold the result.

Pushers Yes

Returns The numeric result of the operation.

Examples

```
tan 45   gives 1.619775
```

Conversions

deg2rad (d2r)

Converts the provided angle from degrees into radians

Purpose

Convert degrees to radians.

Syntax

deg2rad angle -> /result

Arguments

angle

Angle in degrees.

/result

The numeric variable that will hold the result.

Pushers Yes

Returns The numeric result of the operation.

Examples

deg2rad 180 -> /rad (3.14159)

```
deg2rad 180 -> /rad
deg2rad /angle -> /rad
```

rad2deg (r2d)

Converts the provided angle from radians into degrees

Purpose

Convert radians to degrees.

Syntax

rad2deg angle -> /result

Arguments

angle

Angle in radians.

/result

The numeric variable that will hold the result.

Pushers Yes

Returns The numeric result of the operation.

Examples

rad2deg /pi -> /result (180)

```
rad2deg 1 -> /deg
rad2deg /rad -> /deg
```

Note: Radians

radians = (pi * deg) / 180

deg = (180 * radians) / pi

Binary Conversion Operations

b2d

Generate the decimal representation of a binary value

Purpose

The b2d converts the supplied binary number into it's decimal equivalent.

Syntax

b2d value

Arguments

value

A binary value upto 16 bits.

Pushers Yes

Examples

```
b2d 0000000000000001 -> /result     (1)
b2d 0000000010000000 -> /result     (128)
b2d 1111111111111111 -> /result     (65535 )
```

d2b

Generate the binary representation of an integer number

Purpose

The d2b converts the supplied decimal number into it's binary equivalent.

Syntax

d2b value

Arguments

value

A positive integer number. Maximum valid value is 65,535

Pushers` Yes

Examplesv

```
d2b 1 -> /result          (0000000000000001)
d2b 128 -> /result        (0000000010000000)
d2b 65535 -> /result      (1111111111111111)
```

Mapping Numbers

mapi

Map an integer number from one range to another

Purpose

Map re-maps an integer value from one range to another. That is, a value of fromLow would get mapped to toLow, a value of fromHigh to toHigh, values in-between to values in-between, etc.

Syntax

mapi min1 max1 min2 max2 value -> /result

Arguments

min1

The first minimum value

max1

The first maximum value

min2

The second minimum value, to which the first will be mapped.

max2

The second maximum value, to which the first will be mapped.

value

The numeric value to be mapped.

/result

The numeric variable that will hold the result.

Pushers Yes

Returns The numeric result of the operation.

Examples

```
map 0 100 -10 10 25 -> /result
map /min /max /tomin /tomax /value -> /result
```

Notes

Say I have a value of 500 which is normally on a scale of between 0 and 1000, but I want to map it down to a scale of 0 to 100, use this command:

map 0 1000 0 100 500 gives 50

mapf

Map a floating point number from one range to another

Purpose

Map re-maps floating point value from one range to another. That is, a value of fromLow would get mapped to toLow, a value of fromHigh to toHigh, values in-between to values in-between, etc.

Syntax

mapf min1 max1 min2 max2 value -> /result

Arguments

min1

The first minimum value

max1

The first maximum value

min2

The second minimum value, to which the first will be mapped.

max2

The second maximum value, to which the first will be mapped.

value

The numeric value to be mapped.

/result

The numeric variable that will hold the result.

Pushers Yes

Returns The numeric result of the operation.

Examples

```
mapf 0 100 -10 10 25 -> /result
map /min /max /tomin /tomax /value -> /result
```

Notes

Say I have a value of 500 which is normally on a scale of between 0 and 1000, but I want to map it down to a scale of 0 to 100, use this command:

map 0 1000 0 100 500 gives 50

Navigation Commands

compupd

Set the compass unit update interval in secs

Purpose

Set the interval in seconds between compass unit updates. Interval ranges from 10 to 600 secs.

Syntax

compupd interval

Arguments

interval

The update interval in seconds, 10 to 600.

Examples

```
compupd 60
Update compass readings every 60 secs.
```

gpson

Enable the GPS unit

Purpose

Enables the GPS unit

Syntax

gpson

gpsoff

Disable the GPS unit

Purpose

Disables the GPS unit

Syntax

gpsoff

gps

Display the latest readings from the GPS unit

Purpose

Display the latest position and status from the GPS unit.

Syntax

gps

gpschan

Set the channel number that the GPS unit is connected to on the JackBord

Purpose

This command sets the chan no or i/o pin that the GPS unit is connected to. Use the port pin no.

Syntax

gpschan target offset

Arguments

#

The port letter a, b, c, d

~

The port pin number 1 to 5

Examples

```
gpschan a1
GPS unit is attached to port pin A1
```

gpsupd

Set the GPS system update interval in seconds

Purpose

Sets the interval after which a new set of readings will be collected from the attached GPS unit. Default is 5secs

Syntax

gpsupd interval

Arguments

interval

The update interval in seconds from 5 to 600 or 10 minutes.

Pushers No

Examples

```
gpsupd 60
Set update interval to 60secs.
```

sendnav

Send the latest navigation readings via MQTT

Purpose

Sends the latest position and status from the GPS unit via mqtt.

Syntax

sendnav

vnv

Display the navigation status

Purpose

Display the status of the navigation system.

Syntax

vnv

Examples

```
******** Navigation ********
Use GPS      :0
GPS Chan     : 0
gpio pin     : 0
Valid Loc    : 0
GPS Valid    : 0
Sats         : 0
Lat          : 0.000000
Long         : 0.000000
Alt          : 0.000000 m
Course       : 0.000000 deg
Speed        : 0.000000 ms 0.000000 kph

Ages Pos[-1] Alt[-1] Course[-1] Speed[-1]

Date [0] 0/0/2000 Time [0] 00:00:00
```

```
Nav Update Delays Gps [5000]  Compass [5000]

Compass Module On[0] valid [0]  heading [0]deg age [0]  type[0]
```