Package: c64vice (via r-universe)

October 16, 2024

Type Package

Title Interface to Binary Monitor in VICE C64 Emulator

Version 0.1.0

Author mikefc

Maintainer mikefc <mikefc@coolbutuseless.com>

Description Interface to the binary monitor in VICE - the c64 emulator.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Imports R6

Repository https://coolbutuseless.r-universe.dev

RemoteUrl https://github.com/coolbutuseless/c64vice

RemoteRef HEAD

RemoteSha a78dd3ce13b30a6157ede2c0ae8a640d6b570875

Contents

`ByteStream` *Internal helper class for dealing with byte sequences*

Description

Internal helper class for dealing with byte sequences

Internal helper class for dealing with byte sequences

Public fields

`vec` vector of bytes

`idx` current index within self$vec Initialise

Methods

Public methods:

- `ByteStream$new()`
- `ByteStream$advance()`
- `ByteStream$consume()`
- `ByteStream$consume_len2()`

- ByteStream$consume_len4()
- ByteStream$eos()
- ByteStream$clone()

Method `new()`**:**

Usage:

```
ByteStream$new(vec)
```

Arguments:

`vec` vector of bytes. will be cast to integer Advance the indxe pointer without returning the value

Method `advance()`**:**

Usage:

```
ByteStream$advance(i)
```

Arguments:

`i` number of bytes to advance Consume bytes and return the values

Method `consume()`**:**

Usage:

```
ByteStream$consume(n)
```

Arguments:

`n` number of bytes to consume

Method `consume_len2()`**:** Consume 2 bytes and interpret as a little-endian integer

Usage:

```
ByteStream$consume_len2()
```

Method `consume_len4()`**:** Consume 4 bytes and interpret as a little-endian integer

Usage:

```
ByteStream$consume_len4()
```

Method `eos()`**:** Have we reached/exceeded the length of the bytestream

Usage:

```
ByteStream$eos()
```

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
ByteStream$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

machine *c64 machine constants*

Description

c64 machine constants

Usage

```
machine
```

Format

An object of class `list` of length 4.

req *Named list functions to generate raw byte vectors to be sent to VICE*

Description

Functions indexed by both their command code (e.g. "0x31") and by their description (e.g. "registers_get")

Usage

```
req
```

Format

An object of class `list` of length 0.

req_advance_instructions *Step over a certain number of instructions.*

Description

Step over a certain number of instructions.

Usage

```
req_advance_instructions(n, sub1 = TRUE)
```

Arguments

n	number of instructions to jump over
sub1	Should subroutines count as a single instruction?

Value

empty response

---

req_autostart *Load a program then return to the monitor*

Description

Load a program then return to the monitor

Usage

```
req_autostart(filename, file_idx, run_after_load = TRUE)
```

Arguments

filename	Name of file
file_idx	If given a disk image, the index of the file to execute. Default: 0.
run_after_load	logical. Default: TRUE

---

req_banks_available *Gives a listing of all the bank IDs for the running machine with their names.*

Description

Gives a listing of all the bank IDs for the running machine with their names.

Usage

```
req_banks_available()
```

Value

named list of bank IDs

`req_checkpoint_delete` *Deletes any type of checkpoint. (break, watch, trace)*

Description

Deletes any type of checkpoint. (break, watch, trace)

Usage

```
req_checkpoint_delete(checkpoint)
```

Arguments

`checkpoint`	number 32bit int.

Value

empty response

`req_checkpoint_get` *Gets any type of checkpoint. (break, watch, trace)*

Description

Gets any type of checkpoint. (break, watch, trace)

Usage

```
req_checkpoint_get(checkpoint)
```

Arguments

`checkpoint`	number 32bit int.

Value

a named list

checkpoint Checkpoint number

hit Currently hit? logical

start Start address

end End address

stop_when_hit Stop when hit? Logical.

enabled Logical.

cpu_op CPU operatio: 1 = Load, 2 = Store, 4 = Exec

temporary Deletes the checkpoint after it has been hit once. This is similar to "until" command, but it will not resume the emulator.

hit_count Number of hits

ignore_count Ignore count

has_condition Has condition? Logical.

mem_space Memory space.

`req_checkpoint_list` *List of checkpoints*

Description

List of checkpoints

Usage

```
req_checkpoint_list()
```

Value

causes VICE to emit a series of responses (as would be returned by a sequence of `req_checkpoint_get()`) followed by the actual response body which is just a count of the total number of checkpoints

---

`req_checkpoint_set` *Sets any type of checkpoint.*

Description

This combines the functionality of several textual commands (break, watch, # trace) into one, as they are all the same with only minor variations. To set conditions, see section 13.4.8 Condition set (0x22) after executing this one.

Usage

```
req_checkpoint_set(
  start,
  end,
  stop_when_hit = TRUE,
  enabled = TRUE,
  cpu_op = 1,
  temporary = TRUE,
  mem_space = machine$mem_space$main
)
```

Arguments

start	start address
end	end address
stop_when_hit	logical. Default: TRUE
enabled	logical. default: TRUE
cpu_op	1 = load. 2 = store. 4 = exec. devault: 1
temporary	Deletes the checkpoint after it has been hit once. This is similar to "until" command, but it will not resume the emulator.
mem_space	memory space (integer). Default: 0 (main memory). Other acceptable values 1=drive8, 2=drive9, 3=drive10, 4=drive11

Value

checkpoint list. Samve as `req_checkpont_set`

`req_checkpoint_toggle` *Checkpoint toggle*

Description

Checkpoint toggle

Usage

```
req_checkpoint_toggle(checkpoint, enabled)
```

Arguments

checkpoint	number 32bit int.
enabled	Should checkpoint be enabled? logical.

Value

empty response

`req_condition_set` *Sets a condition on an existing checkpoint. It is not currently possible to retrieve conditions after setting them.*

Description

Sets a condition on an existing checkpoint. It is not currently possible to retrieve conditions after setting them.

Usage

```
req_condition_set(checkpoint, condition_expr)
```

Arguments

`checkpoint`	number 32bit int.
`condition_expr`	Condition expression string. This is the same format used on the command line.

Value

empty response

`req_display_get` *Gets the current screen in a requested bit format.*

Description

This function returns a matrix which includes the screen capture including a generous border on each side.

Usage

```
req_display_get()
```

Value

name list of metainformation including 'img' which contains the colour indices at each memory location.

len length of the fields before the display buffer

dwidth, dheight Raw size of the returned matrix of pixels

xoff, yoff Offset within the image to the actual screen pixels

width, height dimensions of screen pixels starting at position (xoff, yoff)

bpp bits per pixel

img raw vector of colour at each pixel.

See Also

[req_palette_get()]

`req_dump` *Saves the machine state to a file.*

Description

Saves the machine state to a file.

Usage

```
req_dump(filename, save_roms = FALSE, save_disks = FALSE)
```

Arguments

`filename`	filename to save machine state into
`save_roms`	save ROMs to snapshot file? Logical. default: FALSE
`save_disks`	save disks to snapshot file. Logical. default: FALSE

Value

empty response

`req_execute_until_return`
Continues execution and returns to the monitor just after the next RTS or RTI is executed.

Description

This command is the same as "return" in the text monitor.

Usage

```
req_execute_until_return()
```

Value

empty response

req_exit *Exit the monitor until the next breakpoint.*

Description

Exit the monitor until the next breakpoint.

Usage

```
req_exit()
```

Value

empty response

req_joyport_set *Set the simulated joyport value.*

Description

Set the simulated joyport value.

Usage

```
req_joyport_set(port, value)
```

Arguments

port	the port to set the value on
value	value to set

Value

empty response

`req_keyboard_feed` *Add text to the keyboard buffer.*

Description

Add text to the keyboard buffer.

Usage

```
req_keyboard_feed(text)
```

Arguments

`text`	PETSCII text

Value

empty response

`req_memory_get` *Reads a chunk of memory from a start address to an end address (inclusive).*

Description

13.4.1 Memory get (0x01)

Usage

```
req_memory_get(start, end, mem_space = 0, side_effects = FALSE, bank_id = 0)
```

Arguments

`start`	start address (integer)
`end`	end address (integer)
`mem_space`	memory space (integer). Default: 0 (main memory). Other acceptable values 1=drive8, 2=drive9, 3=drive10, 4=drive11
`side_effects`	Should the read cause side effects? Default: FALSE
`bank_id`	which bank you want. This is dependent on your machine. If the memspace selected doesn't support banks, this value is ignored.

Details

cmd <- 0x01

Value

vector of raw bytes

See Also

[req_banks_available()]

`req_memory_set` *Writes a chunk of memory from a start address to an end address (inclusive).*

Description

Writes a chunk of memory from a start address to an end address (inclusive).

Usage

```
req_memory_set(
  bytes,
  start,
  bank_id = 0,
  mem_space = machine$mem_space$main,
  side_effects = FALSE
)
```

Arguments

`bytes`	raw vector of bytes to be written to memory
`start`	start address (integer)
`bank_id`	which bank you want. This is dependent on your machine. If the memspace selected doesn't support banks, this value is ignored.
`mem_space`	memory space (integer). Default: 0 (main memory). Other acceptable values 1=drive8, 2=drive9, 3=drive10, 4=drive11
`side_effects`	Should the read cause side effects? Default: FALSE

Value

empty response

req_palette_get *Get the colors in the current palette*

Description

Get the colors in the current palette

Usage

```
req_palette_get()
```

Value

character vector of colours

req_ping *Get an empty response*

Description

Get an empty response

Usage

```
req_ping()
```

Value

empty response

req_quit *Quits VICE.*

Description

Quits VICE.

Usage

```
req_quit()
```

Value

empty response

`req_registers_available` *Gives a listing of all the registers for the running machine with their names.*

Description

Gives a listing of all the registers for the running machine with their names.

Usage

```
req_registers_available(mem_space = machine$mem_space$main)
```

Arguments

`mem_space`	memory space (integer). Default: 0 (main memory). Other acceptable values 1=drive8, 2=drive9, 3=drive10, 4=drive11

Value

named list of regiseters

`req_registers_get` *Get details about the registers*

Description

Get details about the registers

Usage

```
req_registers_get(mem_space = machine$mem_space$main)
```

Arguments

`mem_space`	memory space (integer). Default: 0 (main memory). Other acceptable values 1=drive8, 2=drive9, 3=drive10, 4=drive11

Value

named integer vector where names are the register names.

req_registers_set *Set the register values*

Description

Set the register values

Usage

```
req_registers_set(..., mem_space = machine$mem_space$main)
```

Arguments

...	name/value pairs specifying values for the named registers. e.g. `req_register_set(A = 12)`
mem_space	memory space (integer). Default: 0 (main memory). Other acceptable values 1=drive8, 2=drive9, 3=drive10, 4=drive11

Value

named integer vector where names are the register names.

req_reset *Reset the system or a drive*

Description

Reset the system or a drive

Usage

```
req_reset(hard_reset = TRUE, drive = NULL)
```

Arguments

hard_reset	logical. default: TRUE
drive	integer. 8-11. If set, then this drive is reset rather than the main system.

`req_resource_get` *Get a resource value from the emulator. See section 6.1 Format of resource files.*

Description

Get a resource value from the emulator. See section 6.1 Format of resource files.

Usage

```
req_resource_get(resource)
```

Arguments

`resource`	resource name. See https://vice-emu.sourceforge.io/vice_6.html#SEC84

Value

named list of 'type' of value (string or integer) and 'value'

`req_resource_set` *Set a resource value in the emulator. See section 6.1 Format of resource files.*

Description

Set a resource value in the emulator. See section 6.1 Format of resource files.

Usage

```
req_resource_set(resource, value)
```

Arguments

`resource`	resource name. See https://vice-emu.sourceforge.io/vice_6.html#SEC84
`value`	value to set. allowed: string or integer.

Value

empty response

req_undump *Loads the machine state from a file.*

Description

Loads the machine state from a file.

Usage

```
req_undump(filename)
```

Arguments

filename	filename containing dump of the machine state.

Value

The current program counter position

req_userport_set *Set the simulated userport value.*

Description

Set the simulated userport value.

Usage

```
req_userport_set(value)
```

Arguments

value	value to set

Value

empty response

req_vice_info *Get general information about VICE. Currently returns the versions.*

Description

Get general information about VICE. Currently returns the versions.

Usage

```
req_vice_info()
```

Value

list with vice and SVN versions

run_prg *Run a PRG given as a numeric vector or a filename*

Description

Run a PRG given as a numeric vector or a filename

Usage

```
run_prg(prg, ...)
```

Arguments

prg	filename or raw/numeric vector
...	extra arguments passed to send_req()

save_screenshot *Take a screenshot and save as PNG file*

Description

Take a screenshot and save as PNG file

Usage

```
save_screenshot(filename, scale = 4, keep_border = TRUE)
```

Arguments

filename	PNG filename
scale	scale factor applied to image before saving. defualt: 4
keep_border	keep the borders as part of the screenshot.

send_req *Deliver a request of raw bytes to the VICE binary monitor*

Description

Deliver a request of raw bytes to the VICE binary monitor

Usage

```
send_req(
  request,
  parse_response = TRUE,
  keep_raw = FALSE,
  wait = 0.2,
  host = "localhost",
  port = 6502
)
```

Arguments

request	raw vector
parse_response	default: TRUE. If FALSE then return raw vector
keep_raw	should the raw bytes be returned as part of the parsed response? Default: FALSE
wait	how long to wait before checking for the response to the request. Default: 0.2s. Different commands, networks and Operating Systems will have different requirements here.
host	host to connect to. Default: "localhost"
port	port on host to connect to. Default: 6502. This is the default port set by VICE when running the binary monitor.

Value

named list with parsed response or a raw vector (if `parse_response = FALSE`)

take_screenshot *Take a screenshot*

Description

Take a screenshot

Usage

```
take_screenshot(keep_border = TRUE)
```

Arguments

keep_border	keep the borders as part of the screenshot.

Value

raster

Index