
pysimavr Documentation

Release 0.2.2

ponty

Apr 14, 2017

Contents

1	About	1
2	Basic usage	2
3	Installation	3
3.1	General	3
3.2	Ubuntu 14.04	3
3.3	Uninstall	4
3.4	Fedora 25	4
4	Usage	5
4.1	vcd export example	7
5	File hierarchy	8
6	How to update external sources	9
7	API	10
7.1	low level interface	10
7.2	high level interface	16
	Python Module Index	18

CHAPTER 1

About

pysimavr is a python wrapper for `simavr` which is AVR and arduino simulator

Links:

- home: <https://github.com/ponty/pysimavr>
- documentation: <http://pysimavr.readthedocs.org>
- PYPI: <https://pypi.python.org/pypi/pysimavr>

Features:

- python wrapper using `swig`
- `simavr` source code and `avr-libc` headers are included for easier installation
- object oriented interface on top of the generated interface
- maximum speed can be real-time
- serial communication
- check `simavr` documentation

CHAPTER 2

Basic usage

```
>>> from pysimavr.avr import Avr
>>> avr=Avr(mcu='atmega48',f_cpu=8000000)
>>> firmware = Firmware('lcd.elf')
>>> avr.load_firmware(firmware)
```

```
>>> from pysimavr.sim import ArduinoSim
>>> print ArduinoSim(snippet='Serial.print("hello!");').get_serial()
hello!
```

check [simavr](#) documentation

General

- install `python`
- install `pip`
- optional: install `swig` (for generating code)
- install header files and a static library for Python (for source build)
- install a compiler (for source build)
- install elf library
- install the program:

```
# as root
pip install pysimavr
```

Ubuntu 14.04

```
sudo apt-get install python-pip
sudo apt-get install python-dev gcc libelf-dev arduino
sudo pip install pysimavr
# optional for some tests:
sudo apt-get install freeglut3-dev scons
# optional for generating code:
sudo apt-get install swig
```

Uninstall

```
# as root
pip uninstall pysimavr
```

Fedora 25

```
$ sudo dnf install swig python3-devel libevent-devel elfutils-libelf-devel
$ sudo pip install pysimavr
```

CHAPTER 4

Usage

pysimavr.examples.simple:

```
!-- include('examples/simple.py')--#
from pysimavr.avr import Avr

if __name__ == "__main__":
    avr = Avr(mcu='atmega48', f_cpu=8000000)
    print( avr.pc )
    avr.step(1)
    print( avr.pc )
    avr.step(1)
    print( avr.pc )

    avr.terminate()

!--#
```

Output:

```
!-- sh('python -m pysimavr.examples.simple ')--#
0
2
4
!--#
```

pysimavr.examples.hello:

```
!-- include('examples/hello.py')--#
from pysimavr.sim import ArduinoSim

if __name__ == "__main__":
    s= ArduinoSim(snippet='Serial.println("hello!");').get_serial()
    print(s)

!--#
```

Output:

```
!-- sh('python -m pysimavr.examples.hello ')--#
hello!
```

```
##
```

pysimavr.examples.delay:

```
## include('examples/delay.py')##
from pysimavr.sim import ArduinoSim
import time

snippet = '''
int i=0;
while (1)
{
    Serial.println(i++);
    _delay_ms(1000);
}
'''
t0 = None

def logger(x):
    global t0
    t = time.time()
    if not t0:
        t0 = t
    print t - t0, x

f_cpu=16000000
fps=20
speed=1
timespan=5

if __name__ == "__main__":
    ArduinoSim(snippet=snippet,
               timespan=timespan,
               serial_line_logger=logger,
               f_cpu=f_cpu,
               fps=fps,
               speed=speed,
               ).run()

##
```

Output:

```
## sh('python -m pysimavr.examples.delay ')##
0.0 0

1.00977802277 1

2.01976013184 2

3.02968215942 3

4.03792500496 4

##
```

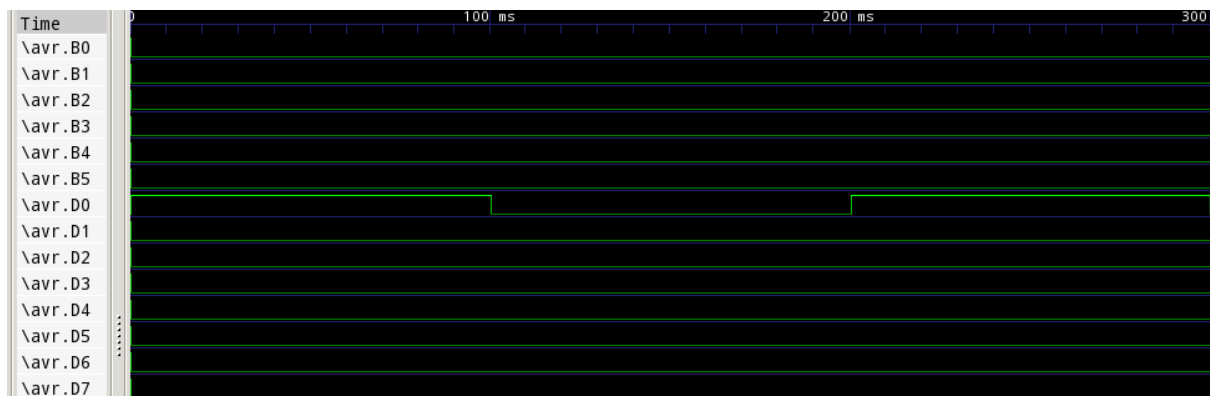

vcd export example

pysimavr.examples.vcd:

```
#!/usr/bin/env python
-- include('examples/vcd.py')--#
from pysimavr.sim import ArduinoSim

vcdfile='delay.vcd'
snippet = '''
    Serial.println("start");
    pinMode(0, OUTPUT);
    digitalWrite(0, HIGH);
    delay(100);
    digitalWrite(0, LOW);
    delay(100);
    digitalWrite(0, HIGH);
    delay(100);
    digitalWrite(0, LOW);
    delay(100);
    Serial.println("end");
'''

if __name__ == "__main__":
    sim = ArduinoSim(snippet=snippet, vcd=vcdfile, timespan=0.5)
    sim.run()
#!/usr/bin/env python
```



CHAPTER 5

File hierarchy

-docs	sphinx documentation
---.build	generated documentation
-pysimavr	main python package, high level classes
---examples	examples
---swig	all swig files (simavr and parts)
-----include	copy of simavr generated *.h files
-----avr	copy from avr-libc
-----parts	some electronic parts in c
-----simavr	simavr as git submodule
-tests	unit tests

CHAPTER 6

How to update external sources

1. copy `avr-libc` headers (Ubuntu folder: `/usr/lib/avr/include/avr/`) into `pysimavr/swig/include/avr`
2. `simavr` is a git submodule. Run 'make' inside `simavr` directory, then copy generated `sim_core_config.h` and `sim_core_decl.h` into `pysimavr/swig/include`

There are 2 interfaces:

- `pysimavr.swig.*`: low level, generated by swig
- `pysimavr.*`: high level classes, they can redirect function calls to low level interface. Example: `Avr` class (high level) has all properties and methods of `avr_t` class (low level) automatically.

low level interface

```
class pysimavr.swig.ac_input.ac_input_t
```

```
    avr
    irq
    value
```

```
class pysimavr.swig.hd44780.hd44780_t
```

```
    avr
    cursor
    datapins
    flags
    h
    irq
    pinstate
    readpins
    vram
    w
```

```
class pysimavr.swig.inverter.inverter_t
```

avr

irq

out

class pysimavr.swig.ledrow.ledrow_t

avr

irq

pinstate

pinstate_changed

class pysimavr.swig.sgm7.sgm7_t

avr

digit_count

digit_pin

digit_port

digit_segments

digit_segments_changed

irq

pinstate

segment_pin

segment_port

class pysimavr.swig.simavr.avr_io_t

avr

dealloc

ioctl

irq

irq_count

irq_ioctl_get

irq_names

kind

next

reset

class pysimavr.swig.simavr.avr_iopin_t

pin

port

class pysimavr.swig.simavr.avr_ioport_external_t

mask

name

value

class pysimavr.swig.simavr.**avr_ioport_getirq_t**

bit

irq

class pysimavr.swig.simavr.**avr_ioport_state_t**

ddr

name

pin

port

class pysimavr.swig.simavr.**avr_ioport_t**

external

io

name

pcint

r_ddr

r_pcint

r_pin

r_port

class pysimavr.swig.simavr.**avr_ioport_t_external**

pull_mask

pull_value

class pysimavr.swig.simavr.**avr_irq_pool_t**

count

irq

class pysimavr.swig.simavr.**avr_irq_t**

flags

hook

irq

name

pool

value

class pysimavr.swig.simavr.**avr_kind_t**

make

names

```
class pysimavr.swig.simavr.avr_regbit_t
```

```
    bit  
    mask  
    reg
```

```
class pysimavr.swig.simavr.avr_symbol_t
```

```
    addr  
    symbol
```

```
class pysimavr.swig.simavr.avr_t
```

```
    address_size  
    aref  
    avcc  
    codeend  
    custom  
    cycle  
    cycle_timers  
    data  
    e2end  
    eind  
    flash  
    flashend  
    frequency  
    fuse  
    gdb  
    gdb_port  
    init  
    interrupt_state  
    interrupts  
    io  
    io_port  
    io_shared_io  
    io_shared_io_count  
    irq_pool  
    log  
    mmcu  
    pc  
    ramend  
    rampz
```

`reset`
`reset_pc`
`run`
`run_cycle_count`
`run_cycle_limit`
`signature`
`sleep`
`sleep_usec`
`sreg`
`state`
`trace`
`trace_data`
`vcc`
`vcd`
`vector_size`

`class pysimavr.swig.simavr.avr_t_custom`

`data`
`deinit`
`init`

`class pysimavr.swig.simavr.avr_t_io`

`irq`
`r`
`w`

`class pysimavr.swig.simavr.avr_t_io_r`

`c`
`param`

`class pysimavr.swig.simavr.avr_t_io_shared_io`

`io`
`used`

`class pysimavr.swig.simavr.avr_t_io_shared_io_io`

`c`
`param`

`class pysimavr.swig.simavr.avr_t_io_w`

`c`
`param`


```
class pysimavr.swig.simavr.avr_trace_data_t
```

```
    codeline
```

```
    old
```

```
    old_pci
```

```
    touched
```

```
class pysimavr.swig.simavr.avr_trace_data_t_old
```

```
    pc
```

```
    sp
```

```
class pysimavr.swig.simavr.avr_vcd_log_t
```

```
    signal
```

```
    value
```

```
    when
```

```
class pysimavr.swig.simavr.avr_vcd_signal_t
```

```
    alias
```

```
    irq
```

```
    name
```

```
    size
```

```
class pysimavr.swig.simavr.avr_vcd_t
```

```
    avr
```

```
    filename
```

```
    log
```

```
    logindex
```

```
    logsize
```

```
    output
```

```
    period
```

```
    signal
```

```
    signal_count
```

```
    start
```

```
class pysimavr.swig.simavr.elf_firmware_t
```

```
    aref
```

```
    avcc
```

```
    bsssize
```

```
    command_register_addr
```

```
    console_register_addr
```

```
    datasize
```

```

eeprom
eesize
external_state
flash
flashbase
flashsize
frequency
mmcu
symbol
symbolcount
trace
tracecount
tracename
traceperiod
vcc

```

```
class pysimavr.swig.simavr.elf_firmware_t_external_state
```

```

mask
port
value

```

```
class pysimavr.swig.simavr.elf_firmware_t_trace
```

```

addr
mask
name

```

high level interface

```
class pysimavr.ac.Ac(avr)
```

```
getirq(pin)
```

```
pysimavr.connect.connect_irqs(irq_out, irq_in, bidirectional=False)
```

```
pysimavr.connect.connect_pins_by_rule(rule, device_map, vcd=None)
```

rule example:

```
B0 -> D4 -> vcd
```

```
B1 <== D5 B2 ==> D6 # B3 <=> D7
```

```
class pysimavr.inverter.Inverter(avr)
```

```
getirq(pin)
```

```
out(i)
```

```
class pysimavr.lcd.Lcd (avr, size=(20, 2))
```

```
    get_char (x, y)
```

```
    getirq (pin)
```

```
    pinstate (pin)
```

```
    reset ()
```

```
class pysimavr.ledrow.LedRow (avr, size=8)
```

```
    getirq (pin)
```

```
    pinstate (i)
```

```
    reset_dirty (i)  
        read and reset
```

```
class pysimavr.sgm7.Sgm7 (avr, size=4)
```

```
    digit_segments (digit_index)
```

```
    getirq (pin)
```

```
    pinindex (pin_name)
```

```
    pinstate (pin)
```

```
    reset_dirty (digit_index)  
        read and reset
```

```
class pysimavr.vcdfile.VcdFile (avr, filename='gtkwave_output.vcd', period=10)
```

```
    add_signal (irq, name=None, bits=1)
```

```
    start ()
```

```
    stop ()
```

```
    terminate ()
```

p

- pysimavr.ac, 16
- pysimavr.connect, 16
- pysimavr.inverter, 16
- pysimavr.lcd, 16
- pysimavr.ledrow, 17
- pysimavr.sgm7, 17
- pysimavr.swig.ac_input, 10
- pysimavr.swig.hd44780, 10
- pysimavr.swig.inverter, 10
- pysimavr.swig.ledrow, 11
- pysimavr.swig.sgm7, 11
- pysimavr.swig.simavr, 11
- pysimavr.vcdfile, 17

A

Ac (class in pysimavr.ac), 16
ac_input_t (class in pysimavr.swig.ac_input), 10
add_signal() (pysimavr.vcdfile.VcdFile method), 17
addr (pysimavr.swig.simavr.avr_symbol_t attribute), 13
addr (pysimavr.swig.simavr.elf_firmware_t_trace attribute), 16
address_size (pysimavr.swig.simavr.avr_t attribute), 13
alias (pysimavr.swig.simavr.avr_vcd_signal_t attribute), 15
aref (pysimavr.swig.simavr.avr_t attribute), 13
aref (pysimavr.swig.simavr.elf_firmware_t attribute), 15
avcc (pysimavr.swig.simavr.avr_t attribute), 13
avcc (pysimavr.swig.simavr.elf_firmware_t attribute), 15
avr (pysimavr.swig.ac_input.ac_input_t attribute), 10
avr (pysimavr.swig.hd44780.hd44780_t attribute), 10
avr (pysimavr.swig.inverter.inverter_t attribute), 10
avr (pysimavr.swig.ledrow.ledrow_t attribute), 11
avr (pysimavr.swig.sgm7.sgm7_t attribute), 11
avr (pysimavr.swig.simavr.avr_io_t attribute), 11
avr (pysimavr.swig.simavr.avr_vcd_t attribute), 15
avr_io_t (class in pysimavr.swig.simavr), 11
avr_iopin_t (class in pysimavr.swig.simavr), 11
avr_ioport_external_t (class in pysimavr.swig.simavr), 11
avr_ioport_getirq_t (class in pysimavr.swig.simavr), 12
avr_ioport_state_t (class in pysimavr.swig.simavr), 12
avr_ioport_t (class in pysimavr.swig.simavr), 12
avr_ioport_t_external (class in pysimavr.swig.simavr), 12
avr_irq_pool_t (class in pysimavr.swig.simavr), 12
avr_irq_t (class in pysimavr.swig.simavr), 12
avr_kind_t (class in pysimavr.swig.simavr), 12
avr_regbit_t (class in pysimavr.swig.simavr), 12
avr_symbol_t (class in pysimavr.swig.simavr), 13
avr_t (class in pysimavr.swig.simavr), 13
avr_t_custom (class in pysimavr.swig.simavr), 14
avr_t_io (class in pysimavr.swig.simavr), 14
avr_t_io_r (class in pysimavr.swig.simavr), 14
avr_t_io_shared_io (class in pysimavr.swig.simavr), 14
avr_t_io_shared_io_io (class in pysimavr.swig.simavr),

14

avr_t_io_w (class in pysimavr.swig.simavr), 14
avr_trace_data_t (class in pysimavr.swig.simavr), 14
avr_trace_data_t_old (class in pysimavr.swig.simavr), 15
avr_vcd_log_t (class in pysimavr.swig.simavr), 15
avr_vcd_signal_t (class in pysimavr.swig.simavr), 15
avr_vcd_t (class in pysimavr.swig.simavr), 15

B

bit (pysimavr.swig.simavr.avr_ioport_getirq_t attribute), 12
bit (pysimavr.swig.simavr.avr_regbit_t attribute), 13
bsssize (pysimavr.swig.simavr.elf_firmware_t attribute), 15

C

c (pysimavr.swig.simavr.avr_t_io_r attribute), 14
c (pysimavr.swig.simavr.avr_t_io_shared_io_io attribute), 14
c (pysimavr.swig.simavr.avr_t_io_w attribute), 14
codeend (pysimavr.swig.simavr.avr_t attribute), 13
codeline (pysimavr.swig.simavr.avr_trace_data_t attribute), 15
command_register_addr (pysimavr.swig.simavr.elf_firmware_t attribute), 15
connect_irqs() (in module pysimavr.connect), 16
connect_pins_by_rule() (in module pysimavr.connect), 16
console_register_addr (pysimavr.swig.simavr.elf_firmware_t attribute), 15
count (pysimavr.swig.simavr.avr_irq_pool_t attribute), 12
cursor (pysimavr.swig.hd44780.hd44780_t attribute), 10
custom (pysimavr.swig.simavr.avr_t attribute), 13
cycle (pysimavr.swig.simavr.avr_t attribute), 13
cycle_timers (pysimavr.swig.simavr.avr_t attribute), 13

D

data (pysimavr.swig.simavr.avr_t attribute), 13

- data (pysimavr.swig.simavr.avr_t_custom attribute), 14
- datapins (pysimavr.swig.hd44780.hd44780_t attribute), 10
- datasize (pysimavr.swig.simavr.elf_firmware_t attribute), 15
- ddr (pysimavr.swig.simavr.avr_ioport_state_t attribute), 12
- dealloc (pysimavr.swig.simavr.avr_io_t attribute), 11
- deinit (pysimavr.swig.simavr.avr_t_custom attribute), 14
- digit_count (pysimavr.swig.sgm7.sgm7_t attribute), 11
- digit_pin (pysimavr.swig.sgm7.sgm7_t attribute), 11
- digit_port (pysimavr.swig.sgm7.sgm7_t attribute), 11
- digit_segments (pysimavr.swig.sgm7.sgm7_t attribute), 11
- digit_segments() (pysimavr.sgm7.Sgm7 method), 17
- digit_segments_changed (pysimavr.swig.sgm7.sgm7_t attribute), 11
- ## E
- e2end (pysimavr.swig.simavr.avr_t attribute), 13
- EEPROM (pysimavr.swig.simavr.elf_firmware_t attribute), 15
- eesize (pysimavr.swig.simavr.elf_firmware_t attribute), 16
- eind (pysimavr.swig.simavr.avr_t attribute), 13
- elf_firmware_t (class in pysimavr.swig.simavr), 15
- elf_firmware_t_external_state (class in pysimavr.swig.simavr), 16
- elf_firmware_t_trace (class in pysimavr.swig.simavr), 16
- external (pysimavr.swig.simavr.avr_ioport_t attribute), 12
- external_state (pysimavr.swig.simavr.elf_firmware_t attribute), 16
- ## F
- filename (pysimavr.swig.simavr.avr_vcd_t attribute), 15
- flags (pysimavr.swig.hd44780.hd44780_t attribute), 10
- flags (pysimavr.swig.simavr.avr_irq_t attribute), 12
- flash (pysimavr.swig.simavr.avr_t attribute), 13
- flash (pysimavr.swig.simavr.elf_firmware_t attribute), 16
- flashbase (pysimavr.swig.simavr.elf_firmware_t attribute), 16
- flashend (pysimavr.swig.simavr.avr_t attribute), 13
- flashsize (pysimavr.swig.simavr.elf_firmware_t attribute), 16
- frequency (pysimavr.swig.simavr.avr_t attribute), 13
- frequency (pysimavr.swig.simavr.elf_firmware_t attribute), 16
- fuse (pysimavr.swig.simavr.avr_t attribute), 13
- ## G
- gdb (pysimavr.swig.simavr.avr_t attribute), 13
- gdb_port (pysimavr.swig.simavr.avr_t attribute), 13
- get_char() (pysimavr.lcd.Lcd method), 17
- getirq() (pysimavr.ac.Ac method), 16
- getirq() (pysimavr.inverter.Inverter method), 16
- getirq() (pysimavr.lcd.Lcd method), 17
- getirq() (pysimavr.ledrow.LedRow method), 17
- getirq() (pysimavr.sgm7.Sgm7 method), 17
- ## H
- h (pysimavr.swig.hd44780.hd44780_t attribute), 10
- hd44780_t (class in pysimavr.swig.hd44780), 10
- hook (pysimavr.swig.simavr.avr_irq_t attribute), 12
- ## I
- init (pysimavr.swig.simavr.avr_t attribute), 13
- init (pysimavr.swig.simavr.avr_t_custom attribute), 14
- interrupt_state (pysimavr.swig.simavr.avr_t attribute), 13
- interrupts (pysimavr.swig.simavr.avr_t attribute), 13
- Inverter (class in pysimavr.inverter), 16
- inverter_t (class in pysimavr.swig.inverter), 10
- io (pysimavr.swig.simavr.avr_ioport_t attribute), 12
- io (pysimavr.swig.simavr.avr_t attribute), 13
- io (pysimavr.swig.simavr.avr_t_io_shared_io attribute), 14
- io_port (pysimavr.swig.simavr.avr_t attribute), 13
- io_shared_io (pysimavr.swig.simavr.avr_t attribute), 13
- io_shared_io_count (pysimavr.swig.simavr.avr_t attribute), 13
- ioctl (pysimavr.swig.simavr.avr_io_t attribute), 11
- irq (pysimavr.swig.ac_input.ac_input_t attribute), 10
- irq (pysimavr.swig.hd44780.hd44780_t attribute), 10
- irq (pysimavr.swig.inverter.inverter_t attribute), 11
- irq (pysimavr.swig.ledrow.ledrow_t attribute), 11
- irq (pysimavr.swig.sgm7.sgm7_t attribute), 11
- irq (pysimavr.swig.simavr.avr_io_t attribute), 11
- irq (pysimavr.swig.simavr.avr_ioport_getirq_t attribute), 12
- irq (pysimavr.swig.simavr.avr_irq_pool_t attribute), 12
- irq (pysimavr.swig.simavr.avr_irq_t attribute), 12
- irq (pysimavr.swig.simavr.avr_t_io attribute), 14
- irq (pysimavr.swig.simavr.avr_vcd_signal_t attribute), 15
- irq_count (pysimavr.swig.simavr.avr_io_t attribute), 11
- irq_ioctl_get (pysimavr.swig.simavr.avr_io_t attribute), 11
- irq_names (pysimavr.swig.simavr.avr_io_t attribute), 11
- irq_pool (pysimavr.swig.simavr.avr_t attribute), 13
- ## K
- kind (pysimavr.swig.simavr.avr_io_t attribute), 11
- ## L
- Lcd (class in pysimavr.lcd), 16
- LedRow (class in pysimavr.ledrow), 17
- ledrow_t (class in pysimavr.swig.ledrow), 11
- log (pysimavr.swig.simavr.avr_t attribute), 13
- log (pysimavr.swig.simavr.avr_vcd_t attribute), 15

logindex (pysimavr.swig.simavr.avr_vcd_t attribute), 15

logsize (pysimavr.swig.simavr.avr_vcd_t attribute), 15

M

make (pysimavr.swig.simavr.avr_kind_t attribute), 12

mask (pysimavr.swig.simavr.avr_ioport_external_t attribute), 11

mask (pysimavr.swig.simavr.avr_regbit_t attribute), 13

mask (pysimavr.swig.simavr.elf_firmware_t_external_state attribute), 16

mask (pysimavr.swig.simavr.elf_firmware_t_trace attribute), 16

mmcu (pysimavr.swig.simavr.avr_t attribute), 13

mmcu (pysimavr.swig.simavr.elf_firmware_t attribute), 16

N

name (pysimavr.swig.simavr.avr_ioport_external_t attribute), 11

name (pysimavr.swig.simavr.avr_ioport_state_t attribute), 12

name (pysimavr.swig.simavr.avr_ioport_t attribute), 12

name (pysimavr.swig.simavr.avr_irq_t attribute), 12

name (pysimavr.swig.simavr.avr_vcd_signal_t attribute), 15

name (pysimavr.swig.simavr.elf_firmware_t_trace attribute), 16

names (pysimavr.swig.simavr.avr_kind_t attribute), 12

next (pysimavr.swig.simavr.avr_io_t attribute), 11

O

old (pysimavr.swig.simavr.avr_trace_data_t attribute), 15

old_pci (pysimavr.swig.simavr.avr_trace_data_t attribute), 15

out (pysimavr.swig.inverter.inverter_t attribute), 11

out() (pysimavr.inverter.Inverter method), 16

output (pysimavr.swig.simavr.avr_vcd_t attribute), 15

P

param (pysimavr.swig.simavr.avr_t_io_r attribute), 14

param (pysimavr.swig.simavr.avr_t_io_shared_io_io attribute), 14

param (pysimavr.swig.simavr.avr_t_io_w attribute), 14

pc (pysimavr.swig.simavr.avr_t attribute), 13

pc (pysimavr.swig.simavr.avr_trace_data_t_old attribute), 15

pcint (pysimavr.swig.simavr.avr_ioport_t attribute), 12

period (pysimavr.swig.simavr.avr_vcd_t attribute), 15

pin (pysimavr.swig.simavr.avr_iopin_t attribute), 11

pin (pysimavr.swig.simavr.avr_ioport_state_t attribute), 12

pinindex() (pysimavr.sgm7.Sgm7 method), 17

pinstate (pysimavr.swig.hd44780.hd44780_t attribute), 10

pinstate (pysimavr.swig.ledrow.ledrow_t attribute), 11

pinstate (pysimavr.swig.sgm7.sgm7_t attribute), 11

pinstate() (pysimavr.lcd.Lcd method), 17

pinstate() (pysimavr.ledrow.LedRow method), 17

pinstate() (pysimavr.sgm7.Sgm7 method), 17

pinstate_changed (pysimavr.swig.ledrow.ledrow_t attribute), 11

pool (pysimavr.swig.simavr.avr_irq_t attribute), 12

port (pysimavr.swig.simavr.avr_iopin_t attribute), 11

port (pysimavr.swig.simavr.avr_ioport_state_t attribute), 12

port (pysimavr.swig.simavr.elf_firmware_t_external_state attribute), 16

pull_mask (pysimavr.swig.simavr.avr_ioport_t_external attribute), 12

pull_value (pysimavr.swig.simavr.avr_ioport_t_external attribute), 12

pysimavr.ac (module), 16

pysimavr.connect (module), 16

pysimavr.inverter (module), 16

pysimavr.lcd (module), 16

pysimavr.ledrow (module), 17

pysimavr.sgm7 (module), 17

pysimavr.swig.ac_input (module), 10

pysimavr.swig.hd44780 (module), 10

pysimavr.swig.inverter (module), 10

pysimavr.swig.ledrow (module), 11

pysimavr.swig.sgm7 (module), 11

pysimavr.swig.simavr (module), 11

pysimavr.vcdfile (module), 17

R

r (pysimavr.swig.simavr.avr_t_io attribute), 14

r_ddr (pysimavr.swig.simavr.avr_ioport_t attribute), 12

r_pcint (pysimavr.swig.simavr.avr_ioport_t attribute), 12

r_pin (pysimavr.swig.simavr.avr_ioport_t attribute), 12

r_port (pysimavr.swig.simavr.avr_ioport_t attribute), 12

ramend (pysimavr.swig.simavr.avr_t attribute), 13

rampz (pysimavr.swig.simavr.avr_t attribute), 13

readpins (pysimavr.swig.hd44780.hd44780_t attribute), 10

reg (pysimavr.swig.simavr.avr_regbit_t attribute), 13

reset (pysimavr.swig.simavr.avr_io_t attribute), 11

reset (pysimavr.swig.simavr.avr_t attribute), 13

reset() (pysimavr.lcd.Lcd method), 17

reset_dirty() (pysimavr.ledrow.LedRow method), 17

reset_dirty() (pysimavr.sgm7.Sgm7 method), 17

reset_pc (pysimavr.swig.simavr.avr_t attribute), 14

run (pysimavr.swig.simavr.avr_t attribute), 14

run_cycle_count (pysimavr.swig.simavr.avr_t attribute), 14

run_cycle_limit (pysimavr.swig.simavr.avr_t attribute), 14

S

segment_pin (pysimavr.swig.sgm7.sgm7_t attribute), 11

segment_port (pysimavr.swig.sgm7.sgm7_t attribute), 11

Sgm7 (class in pysimavr.sgm7), 17
 sgm7_t (class in pysimavr.swig.sgm7), 11
 signal (pysimavr.swig.simavr.avr_vcd_log_t attribute), 15
 signal (pysimavr.swig.simavr.avr_vcd_t attribute), 15
 signal_count (pysimavr.swig.simavr.avr_vcd_t attribute), 15
 signature (pysimavr.swig.simavr.avr_t attribute), 14
 size (pysimavr.swig.simavr.avr_vcd_signal_t attribute), 15
 sleep (pysimavr.swig.simavr.avr_t attribute), 14
 sleep_usec (pysimavr.swig.simavr.avr_t attribute), 14
 sp (pysimavr.swig.simavr.avr_trace_data_t_old attribute), 15
 sreg (pysimavr.swig.simavr.avr_t attribute), 14
 start (pysimavr.swig.simavr.avr_vcd_t attribute), 15
 start() (pysimavr.vcdfile.VcdFile method), 17
 state (pysimavr.swig.simavr.avr_t attribute), 14
 stop() (pysimavr.vcdfile.VcdFile method), 17
 symbol (pysimavr.swig.simavr.avr_symbol_t attribute), 13
 symbol (pysimavr.swig.simavr.elf_firmware_t attribute), 16
 symbolcount (pysimavr.swig.simavr.elf_firmware_t attribute), 16

T

terminate() (pysimavr.vcdfile.VcdFile method), 17
 touched (pysimavr.swig.simavr.avr_trace_data_t attribute), 15
 trace (pysimavr.swig.simavr.avr_t attribute), 14
 trace (pysimavr.swig.simavr.elf_firmware_t attribute), 16
 trace_data (pysimavr.swig.simavr.avr_t attribute), 14
 tracecount (pysimavr.swig.simavr.elf_firmware_t attribute), 16
 tracename (pysimavr.swig.simavr.elf_firmware_t attribute), 16
 traceperiod (pysimavr.swig.simavr.elf_firmware_t attribute), 16

U

used (pysimavr.swig.simavr.avr_t_io_shared_io attribute), 14

V

value (pysimavr.swig.ac_input.ac_input_t attribute), 10
 value (pysimavr.swig.simavr.avr_ioport_external_t attribute), 11
 value (pysimavr.swig.simavr.avr_irq_t attribute), 12
 value (pysimavr.swig.simavr.avr_vcd_log_t attribute), 15
 value (pysimavr.swig.simavr.elf_firmware_t_external_state attribute), 16
 vcc (pysimavr.swig.simavr.avr_t attribute), 14
 vcc (pysimavr.swig.simavr.elf_firmware_t attribute), 16
 vcd (pysimavr.swig.simavr.avr_t attribute), 14
 VcdFile (class in pysimavr.vcdfile), 17

vector_size (pysimavr.swig.simavr.avr_t attribute), 14
 vram (pysimavr.swig.hd44780.hd44780_t attribute), 10

W

w (pysimavr.swig.hd44780.hd44780_t attribute), 10
 w (pysimavr.swig.simavr.avr_t_io attribute), 14
 when (pysimavr.swig.simavr.avr_vcd_log_t attribute), 15