softusbduino Documentation

Release 1.1.0

ponty

March 04, 2012

CONTENTS

softusbduino

Date March 04, 2012

PDF softusbduino.pdf

CHAPTER

ONE

ABOUT

softusbduino is a Python package and Arduino firmware library. They can be used together to control the Arduino board over USB in Python.

Links:

- home: https://github.com/ponty/softusbduino
- documentation: http://ponty.github.com/softusbduino

Hierarchy: Python Application -> softusbduino python library -> PyUSB -> libusb -> USB cable -> V-USB hardware -> Arduino -> V-USB library -> softusbduino firmware

Features:

- Possible usage: prototyping or creating simple low speed USB devices.
- firmware should be load only once to the Arduino board.
- 1 low level call takes 2 ms in tests
- **python library functions:**
  - read or write all registers
  - call arduino functions
  - read many defines (example: F_CPU)
- Python USB back-end: PyUSB 1.0 library
- Arduino USB back-end: V-USB library

Known problems:

- tested only on Linux + arduino 0022 + ATmega88 board
- pull-up read is not implemented
- PWM read is not implemented
- PWM config is hardcoded

similar projects:

- https://github.com/HashNuke/Python-Arduino-Prototyping-API
- http://code.google.com/p/vusb-for-arduino/
- http://code.google.com/p/pyduino/

CHAPTER

TWO

BASIC USAGE OF PROTOTYPING

```
from softusbduino.protoapi import *

def setup():
    pinMode(13, OUTPUT);

def loop():
    digitalWrite(13, HIGH);
    delay(1000);
    digitalWrite(13, LOW);
    delay(1000);

sketch = Sketch(setup, loop)
sketch.run()
```

CHAPTER

THREE

INSTALLATION

3.1 General

- install Python
- install pip
- install arduino
- install libusb
- **install SoftUsb subdirectory as arduino library**
  - Manual installation: http://arduino.cc/en/Guide/Environment#libraries
  - **Automatic installation:**
    - install confduino
    - install the library: `python -m confduino.libinstall https://github.com/ponty/softusbduino/zipball/master`
- install python package:

  ```
  # as root
  pip install https://github.com/ponty/softusbduino/zipball/master
  ```

3.2 Ubuntu

```
sudo apt-get install arduino python-pip libusb-1.0-0
sudo pip install confduino
sudo pip install https://github.com/ponty/softusbduino/zipball/master
sudo python -m confduino.libinstall https://github.com/ponty/softusbduino/zipball/master
# optional for examples
sudo pip install matplotlib traits traitsui
```

3.3 Upload firmware

1. start Arduino
2. open examples > SoftUsb > Simple
3. upload to board

CHAPTER

FOUR

USAGE

```
>>> from softusbduino import *
>>>
>>> mcu = Arduino()
>>>
>>> # reset pin directions
>>> mcu.reset()
>>>
>>> # constants in python library
>>> print '0x%X' % mcu.usb.id_vendor
0x16C0
>>> print '0x%X' % mcu.usb.id_product
0x5DF
>>> print mcu.bandgap_voltage
1.1
>>>
>>> # constants in firmware
>>> print mcu.pins.usb_minus_pin
4
>>> print mcu.pins.usb_plus_pin
2
>>> print mcu.pins.count
20
>>> print mcu.pins.count_digital
14
>>> print mcu.pins.count_analog
6
>>> print mcu.pins.range_all
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
>>> print mcu.pins.range_digital
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
>>> print mcu.pins.range_analog
[14, 15, 16, 17, 18, 19]
>>>
>>> # supply voltage
>>> print mcu.vcc.voltage
4.87619047619
>>> print mcu.vcc.u_voltage
4.87619047619+/-0.042218099361
>>>
>>> # pin
>>> print mcu.pin(8).nr
8
>>> print mcu.pin('D8').nr
8
>>> print mcu.pin('A2').nr
16
>>> print mcu.pin('D13').programming_function
SCK
```

```
>>>
>>> # pin mode
>>> mcu.pins.write_mode(8, OUTPUT)
>>> print mcu.pins.read_mode(8)
1
>>> print mcu.pin(’D8’).read_mode()
1
>>> print mcu.pin(’D8’).mode
1
>>> mcu.pin(’D8’).mode = INPUT
>>> print mcu.pins.read_mode(8)
0
>>>
>>> # analog read
>>> print mcu.pins.read_analog(15)
306
>>> print mcu.pin(’A2’).read_analog()
343
>>> print mcu.pin(’A2’).analog
332
>>>
>>> # digital read
>>> print mcu.pins.read_digital(8)
1
>>> print mcu.pin(’D8’).read_digital()
1
>>> print mcu.pin(’D8’).digital
1
>>>
>>> # pullup
>>> mcu.pins.write_pullup(8, HIGH)
>>> mcu.pin(’D8’).write_pullup(HIGH)
>>>
>>> # digital write
>>> mcu.pins.write_mode(8, OUTPUT)
>>> mcu.pins.write_digital(8, HIGH)
>>> mcu.pin(’D8’).write_digital(HIGH)
>>> mcu.pin(’D8’).digital = HIGH
>>>
>>> # PWM
>>> print mcu.pin(’D9’).pwm.available
True
>>> print mcu.pin(’D9’).pwm.timer_register_name
TCCR1B
>>> print mcu.pin(’D9’).pwm.frequencies_available
[39062.5, 4882.8125, 610.3515625, 152.587890625, 38.14697265625]
>>> print mcu.pin(’D9’).pwm.frequency
610.3515625
>>> print mcu.pin(’D9’).pwm.divisors_available
[1, 8, 64, 256, 1024]
>>> print mcu.pin(’D9’).pwm.divisor
64
>>> mcu.pin(’D9’).pwm.divisor = 256
>>> print mcu.pin(’D9’).pwm.frequency
152.587890625
>>> print mcu.pin(’D9’).pwm.divisor
256
>>> mcu.pin(’D9’).pwm.frequency = 38
>>> print mcu.pin(’D9’).pwm.frequency
38.1469726562
>>> print mcu.pin(’D9’).pwm.divisor
1024
>>> mcu.pins.pwm.write_value(9, 54)
>>> mcu.pin(’D9’).pwm.write_value(44)
>>> mcu.pin(’D9’).pwm.value = 34
>>>
>>> # read defines
>>> print mcu.define(’F_CPU’)
20000000
>>> print mcu.defines.value(’F_CPU’)
20000000
>>> print mcu.defines.exists(’F_CPU’)
True
>>>
>>> print mcu.define(’MCU_DEFINED’)
__AVR_ATmega88__
>>> print mcu.define(’F_CPU’)
20000000
>>> print mcu.define(’__DATE__’)
Mar  1 2012
>>> print mcu.define(’MOSI’)
11
>>> print mcu.define(’USB_CFG_DMINUS_BIT’)
4
>>> print mcu.define(’ARDUINO’)
22
>>> print mcu.define(’__AVR_LIBC_VERSION__’)
10701
>>> print mcu.define(’A0’)
14
>>>
>>> # read/write register
>>> mcu.register(’DDRB’).value = 0
>>> print mcu.registers.read_value(’DDRB’)
0
>>> print mcu.register(’DDRB’).read_value()
0
>>> print mcu.register(’DDRB’).value
0
>>> print mcu.pin(8).mode
0
>>> mcu.register(’DDRB’).value = 1
>>> print mcu.register(’DDRB’).value
1
>>> print mcu.pin(8).mode
1
>>> mcu.pin(8).mode = INPUT
>>> print mcu.register(’DDRB’).value
0
>>> print mcu.pin(8).mode
0
>>>
>>>
>>> mcu.reset()
```

4.1 Code generation

Integer defines should be listed in softusbduino/intdefs.csv. String defines are hardcoded. Registers and MCU names are read from AVR Libc directory (/usr/lib/avr/include/avr/).

Run codegen.py to update generated files:

- softusbduino/generated_registers.csv

- SoftUsb/generated_registers.h
- SoftUsb/generated_intdefs.h
- SoftUsb/generated_mcu.h
- SoftUsb/generated_version.h

CHAPTER

FIVE

EXAMPLES

5.1 Simple example

```
from entrypoint2 import entrypoint
from softusbduino.arduino import Arduino


@entrypoint
def main():
    mcu = Arduino()
    print 'F_CPU=', mcu.define('F_CPU')
    print 'DDRC=', mcu.register('DDRC').read_value()
```

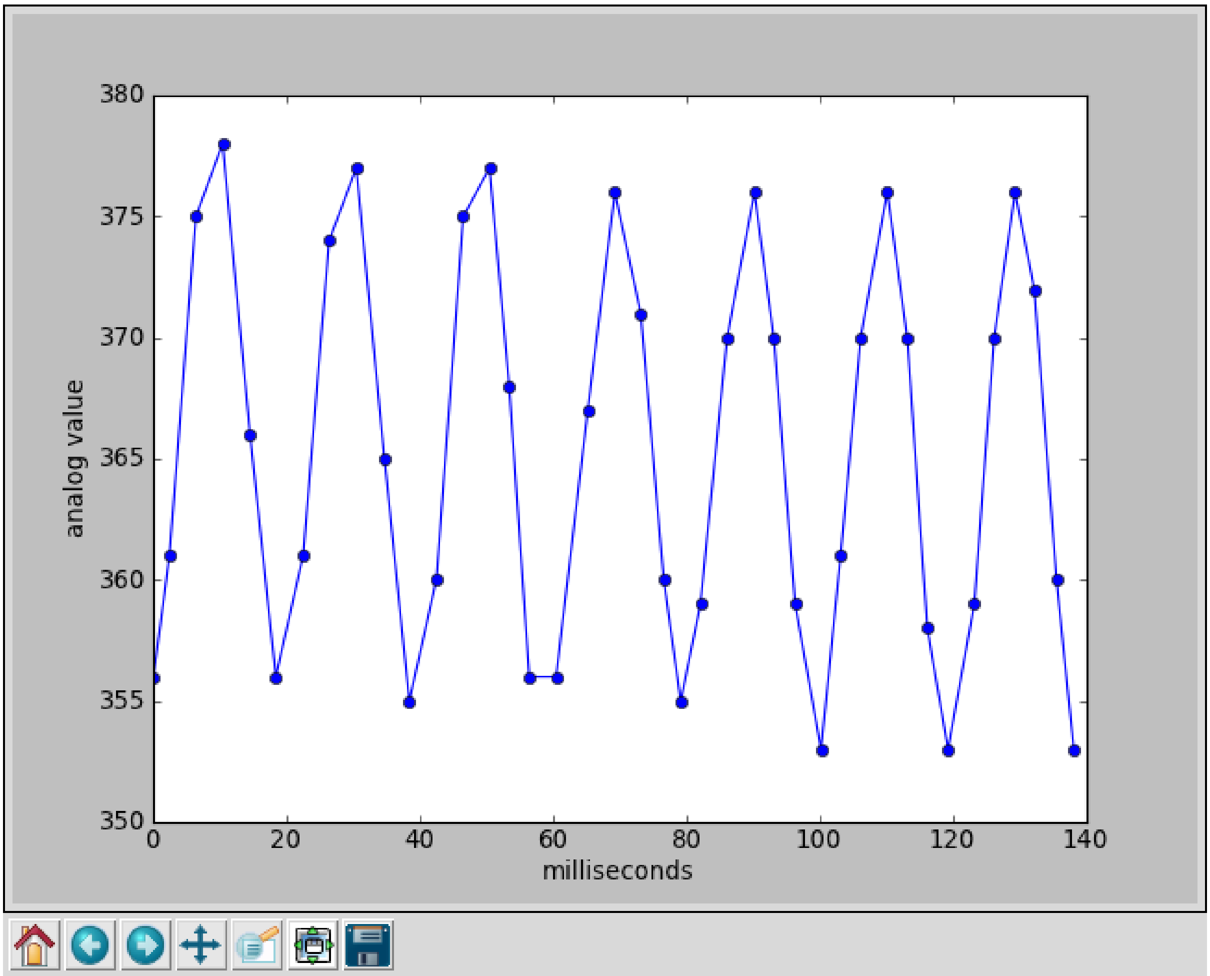
```
$ python -m softusbduino.examples.simple
F_CPU= 20000000
DDRC= 0
```

5.2 Plot

```
from entrypoint2 import entrypoint
from matplotlib.ticker import FuncFormatter
from softusbduino.arduino import Arduino
import matplotlib.pyplot as plt
import time

@entrypoint
def main(n=40, pin_nr=13, reset=False):
    '''
    measuring analog input
    '''
    mcu = Arduino(reset=reset)
    pin = mcu.pin(pin_nr)

    x = []
    y = []
    start = time.time()
    for i in range(n):
        t = time.time() - start
        v = pin.read_analog()
        x.append(t)
        y.append(v)
    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.plot(x, y, 'b-o')
    ax.yaxis.set_major_formatter(FuncFormatter(lambda x, pos: ('%d') % (x)))
    ax.set_ylabel('analog value')

    ax.xaxis.set_major_formatter(FuncFormatter(lambda x, pos: '%.0f' % (1000 * x)))
    ax.set_xlabel('milliseconds')
    plt.show()
```

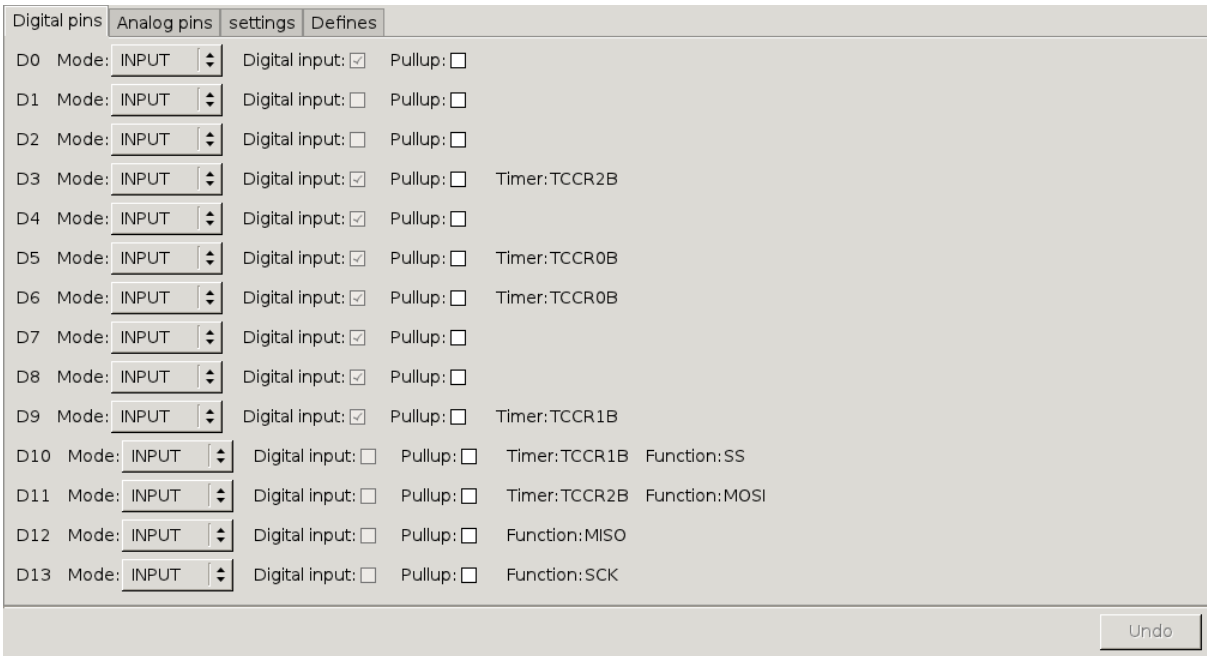
```
$ python -m softusbduino.examples.analogplot --help
usage: analogplot.py [-h] [--n N] [-p PIN_NR] [-r] [--debug]

measuring analog input

optional arguments:
  -h, --help            show this help message and exit
  --n N
  -p PIN_NR, --pin-nr PIN_NR
  -r, --reset
  --debug               set logging level to DEBUG
```

```
$ python -m softusbduino.examples.analogplot
```



5.3 Demo GUI

```
$ python -m softusbduino.examples.guidemo
```



D0 Mode: INPUT Digital input: ☑ Pullup: □
D1 Mode: INPUT Digital input: □ Pullup: □
D2 Mode: INPUT Digital input: □ Pullup: □
D3 Mode: INPUT Digital input: ☑ Pullup: □ Timer: TCCR2B
D4 Mode: INPUT Digital input: ☑ Pullup: □
D5 Mode: INPUT Digital input: ☑ Pullup: □ Timer: TCCR0B
D6 Mode: INPUT Digital input: ☑ Pullup: □ Timer: TCCR0B
D7 Mode: INPUT Digital input: ☑ Pullup: □
D8 Mode: INPUT Digital input: ☑ Pullup: □
D9 Mode: INPUT Digital input: ☑ Pullup: □ Timer: TCCR1B
D10 Mode: INPUT Digital input: □ Pullup: □ Timer: TCCR1B Function: SS
D11 Mode: INPUT Digital input: □ Pullup: □ Timer: TCCR2B Function: MOSI
D12 Mode: INPUT Digital input: □ Pullup: □ Function: MISO
D13 Mode: INPUT Digital input: □ Pullup: □ Function: SCK

5.4 prototyping

softusbduino/examples/proto/Blink.py

```
'''
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  Converted from Arduino example.
'''

from softusbduino.protoapi import *

def setup():
    pinMode(13, OUTPUT);

def loop():
    digitalWrite(13, HIGH);
    delay(1000);
    digitalWrite(13, LOW);
    delay(1000);



sketch = Sketch(setup, loop)
sketch.run()
```

softusbduino/examples/proto/AnalogInOutSerial.py

```
'''
  Analog input, analog output, serial output

 Reads an analog input pin, maps the result to a range from 0 to 255
 and uses the result to set the pulsewidth modulation (PWM) of an output pin.
 Also prints the results to the serial monitor.

 The circuit:
 * potentiometer connected to analog pin 0.
```

```
   Center pin of the potentiometer goes to the analog pin.
   side pins of the potentiometer go to +5V and ground
 * LED connected from digital pin 9 to ground

  Converted from Arduino example.
'''

from softusbduino.protoapi import *

# These constants won't change.  They're used to give names
# to the pins used:
analogInPin = A0;  # Analog input pin that the potentiometer is attached to
analogOutPin = 9; # Analog output pin that the LED is attached to

sensorValue = 0;        # value read from the pot
outputValue = 0;        # value output to the PWM (analog out)

def setup():
    # initialize serial communications at 9600 bps:
    Serial_begin(9600);


def loop():
    # read the analog in value:
    sensorValue = analogRead(analogInPin);
    # map it to the range of the analog out:
    outputValue = map(sensorValue, 0, 1023, 0, 255);
    # change the analog out value:
    analogWrite(analogOutPin, outputValue);

    # print the results to the serial monitor:
    Serial_print("sensor = " );
    Serial_print(sensorValue);
    Serial_print("\t output = ");
    Serial_println(outputValue);

    # wait 10 milliseconds before the next loop
    # for the analog-to-digital converter to settle
    # after the last reading:
    delay(10);

sketch = Sketch(setup, loop)
sketch.run()
```

CHAPTER

SIX

1WIRE EXAMPLES

6.1 Reading temperature

```
from entrypoint2 import entrypoint
from softusbduino.arduino import Arduino
import time


@entrypoint
def main(
         pin=’D9’,
         timeout=10,
         ):
    mcu = Arduino()
    bus = mcu.bus1wire(pin)
    devs = bus.search()
    d = devs[0]
    print ’address=’, d.address_str
    print ’address_valid=’, d.address_valid
    print ’chip=’, d.chip
    print ’resolution=’, d.resolution, ’bit’

    start = time.time()
    while 1:
        x = d.scratchpad()
        TEMPL = ’{t}  T={x.celsius:>10} C , resolution={x.resolution} , connected={x.connected} da
        print TEMPL.format(x=x,
                           t=time.ctime(x.t),
                           )
        time.sleep(0.1)
        if timeout > 0:
            if timeout < time.time() - start:
                break
```

```
$ python -m softusbduino.examples.onewire_demo
address= 28.89.E1.2E.03.00.00.F4
address_valid= True
chip= DS18B20
resolution= 12 bit
Sun Mar  4 21:25:27 2012  T=      19.5 C , resolution=12 , connected=True data=[56, 1, 75, 70, 12
Sun Mar  4 21:25:30 2012  T=   19.5625 C , resolution=12 , connected=True data=[57, 1, 75, 70, 12
Sun Mar  4 21:25:33 2012  T=   19.5625 C , resolution=12 , connected=True data=[57, 1, 75, 70, 12
Sun Mar  4 21:25:37 2012  T=   19.5625 C , resolution=12 , connected=True data=[57, 1, 75, 70, 12
```

CHAPTER

SEVEN

TESTS

Test system versions:

```
$ python -m softusbduino.lsversion
platform        Linux-3.0.0-16-generic-i686-athlon-with-LinuxMint-12-lisa
python          2.7.2+
```

Performance test:

```
$ python -m softusbduino.check.performance
performance test
n= 100

pins.read_analog(0)                     3.57 ms per call,   280 call per second
pins.write_mode(8,0)                    3.57 ms per call,   280 call per second
pins.read_digital(8)                    3.26 ms per call,   307 call per second
defines.value("__TIME__")               0.12 ms per call,  8306 call per second
defines.exists("__TIME__")              0.07 ms per call, 15230 call per second
defines.exists("xx")                    0.07 ms per call, 14428 call per second
define("A0")                            0.07 ms per call, 14265 call per second
registers.read_value("DDRB")            3.76 ms per call,   266 call per second
registers.exists("DDRB")                0.07 ms per call, 13897 call per second
registers.exists("xx")                  0.08 ms per call, 13050 call per second
register("DDRB").value                  3.29 ms per call,   304 call per second
register("DDRB").read_value()           3.40 ms per call,   294 call per second
register("DDRB").exists                 0.07 ms per call, 14152 call per second
register("xx").exists                   0.08 ms per call, 12739 call per second
vcc.voltage                             0.43 ms per call,  2330 call per second
vcc.read_voltage()                     21.31 ms per call,    47 call per second
read_vcc()                             21.49 ms per call,    47 call per second
pins.count                              0.16 ms per call,  6402 call per second
pins.usb_minus_pin                      0.11 ms per call,  8928 call per second
pins.usb_plus_pin                       0.15 ms per call,  6496 call per second
firmware_test()                         0.08 ms per call, 12167 call per second
pins.read_mode(0)                       3.52 ms per call,   284 call per second
reset()                               121.72 ms per call,     8 call per second
```

Dump state:

```
$ python -m softusbduino.check.dump

================================
Arduino() attributes:
================================
Rout            =              15
adc_accuracy    =               2
analog_range    =       (0, 1023)
bandgap_voltage =             1.1
defines         = <softusbduino.defines.Defines object at 0x874672c>
lowlevel_1wire  = <softusbduino.onewire.LowLevel1Wire object at 0x874a8ec>
```

```
lowlevel_defines = <softusbduino.defines.DefinesLowLevel object at 0x87466cc>
lowlevel_delaytest = <softusbduino.delaytest.DelayTestLowLevel object at 0x874a96c>
lowlevel_pins   = <softusbduino.pin.PinsLowLevel object at 0xb78193ec>
lowlevel_pwm    = <softusbduino.pwmpin.PwmLowLevel object at 0x874a9cc>
lowlevel_registers = <softusbduino.registers.RegistersLowLevel object at 0x874aa4c>
pins            = <softusbduino.pin.Pins object at 0x8746bcc>
pwm             = <softusbduino.pwmpin.Pwm object at 0x874aacc>
registers       = <softusbduino.registers.Registers object at 0x874ab4c>
serializer      = <softusbduino.ser.Serializer object at 0x874666c>
usb             = <softusbduino.usbdevice.UsbDevice object at 0x86ed62c>
vcc             = <softusbduino.vcc.Vcc object at 0x874ad0c>

================================
Arduino().pins attributes:
================================
base            = <softusbduino.pin.PinsLowLevel object at 0xb78193ec>
count           =              20
count_analog    =               6
count_digital   =              14
defines         = <softusbduino.defines.Defines object at 0x874672c>
mcu             = <softusbduino.arduino.Arduino object at 0x868514c>
range_all       = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
range_analog    = [14, 15, 16, 17, 18, 19]
range_digital   = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
usb_minus_pin   =               4
usb_plus_pin    =               2

================================
Arduino().pin(nr) attributes:
================================
---------- nr=0 ---------------
A0              =              14
analog          =             342
analog_obj      = <softusbduino.pin.AnalogInputValue object at 0x87622cc>
base            = <softusbduino.pin.Pins object at 0x8746bcc>
digital         =               1
digital_in      =               1
digital_out     =            None
is_analog       =           False
is_digital      =            True
is_usb_minus    =           False
is_usb_plus     =           False
mcu             = <softusbduino.arduino.Arduino object at 0x868514c>
mode            =               0
name            =              D0
nr              =               0
nr_analog       =            None
programming_function =            None
pwm             = <softusbduino.pwmpin.PwmPin object at 0x876240c>
---------- nr=1 ---------------
A0              =              14
analog          =             342
analog_obj      = <softusbduino.pin.AnalogInputValue object at 0x876248c>
base            = <softusbduino.pin.Pins object at 0x8746bcc>
digital         =               0
digital_in      =               0
digital_out     =            None
is_analog       =           False
is_digital      =            True
is_usb_minus    =           False
is_usb_plus     =           False
mcu             = <softusbduino.arduino.Arduino object at 0x868514c>
mode            =               0
```

```
name            =              D1
nr              =               1
nr_analog       =            None
programming_function =            None
pwm             = <softusbduino.pwmpin.PwmPin object at 0x87624ec>
---------- nr=2 ---------------
A0              =              14
analog          =             347
analog_obj      = <softusbduino.pin.AnalogInputValue object at 0x876254c>
base            = <softusbduino.pin.Pins object at 0x8746bcc>
digital         =               0
digital_in      =               0
digital_out     =            None
is_analog       =           False
is_digital      =            True
is_usb_minus    =           False
is_usb_plus     =            True
mcu             = <softusbduino.arduino.Arduino object at 0x868514c>
mode            =               0
name            =              D2
nr              =               2
nr_analog       =            None
programming_function =            None
pwm             = <softusbduino.pwmpin.PwmPin object at 0x876256c>
---------- nr=3 ---------------
A0              =              14
analog          =             368
analog_obj      = <softusbduino.pin.AnalogInputValue object at 0x87625ec>
base            = <softusbduino.pin.Pins object at 0x8746bcc>
digital         =               1
digital_in      =               1
digital_out     =            None
is_analog       =           False
is_digital      =            True
is_usb_minus    =           False
is_usb_plus     =           False
mcu             = <softusbduino.arduino.Arduino object at 0x868514c>
mode            =               0
name            =              D3
nr              =               3
nr_analog       =            None
programming_function =            None
pwm             = <softusbduino.pwmpin.PwmPin object at 0x87625ec>
--- pwm ---
available       =            True
base            = <softusbduino.pwmpin.Pwm object at 0x874aacc>
base_divisor    =             512
divisor         =              64
divisors_available = [1, 8, 32, 64, 128, 256, 1024]
frequencies_available = [39062.5, 4882.8125, 1220.703125, 610.3515625, 305.17578125, 152.587890625
frequency       =     610.3515625
pin             = <softusbduino.pin.Pin object at 0x87621ec>
timer_mode      =               4
timer_register_name =          TCCR2B
---------- nr=4 ---------------
A0              =              14
analog          =             350
analog_obj      = <softusbduino.pin.AnalogInputValue object at 0x87626cc>
base            = <softusbduino.pin.Pins object at 0x8746bcc>
digital         =               1
digital_in      =               1
digital_out     =            None
is_analog       =           False
```

```
is_digital      =            True
is_usb_minus    =            True
is_usb_plus     =           False
mcu             = <softusbduino.arduino.Arduino object at 0x868514c>
mode            =               0
name            =              D4
nr              =               4
nr_analog       =            None
programming_function =            None
pwm             = <softusbduino.pwmpin.PwmPin object at 0x87626ec>
---------- nr=5 ---------------
A0              =              14
analog          =             353
analog_obj      = <softusbduino.pin.AnalogInputValue object at 0x876276c>
base            = <softusbduino.pin.Pins object at 0x8746bcc>
digital         =               0
digital_in      =               0
digital_out     =            None
is_analog       =           False
is_digital      =            True
is_usb_minus    =           False
is_usb_plus     =           False
mcu             = <softusbduino.arduino.Arduino object at 0x868514c>
mode            =               0
name            =              D5
nr              =               5
nr_analog       =            None
programming_function =            None
pwm             = <softusbduino.pwmpin.PwmPin object at 0x87627ac>
--- pwm ---
available       =            True
base            = <softusbduino.pwmpin.Pwm object at 0x874aacc>
base_divisor    =             256
divisor         =              64
divisors_available = [1, 8, 64, 256, 1024]
frequencies_available = [78125.0, 9765.625, 1220.703125, 305.17578125, 76.2939453125]
frequency       =     1220.703125
pin             = <softusbduino.pin.Pin object at 0x87624cc>
timer_mode      =               3
timer_register_name =          TCCR0B
---------- nr=6 ---------------
A0              =              14
analog          =             368
analog_obj      = <softusbduino.pin.AnalogInputValue object at 0x874a2ec>
base            = <softusbduino.pin.Pins object at 0x8746bcc>
digital         =               0
digital_in      =               0
digital_out     =            None
is_analog       =           False
is_digital      =            True
is_usb_minus    =           False
is_usb_plus     =           False
mcu             = <softusbduino.arduino.Arduino object at 0x868514c>
mode            =               0
name            =              D6
nr              =               6
nr_analog       =            None
programming_function =            None
pwm             = <softusbduino.pwmpin.PwmPin object at 0x87628ac>
--- pwm ---
available       =            True
base            = <softusbduino.pwmpin.Pwm object at 0x874aacc>
base_divisor    =             256
```

```
divisor         =              64
divisors_available = [1, 8, 64, 256, 1024]
frequencies_available = [78125.0, 9765.625, 1220.703125, 305.17578125, 76.2939453125]
frequency       =     1220.703125
pin             = <softusbduino.pin.Pin object at 0x87626ac>
timer_mode      =               3
timer_register_name =          TCCR0B
---------- nr=7 ---------------
A0              =              14
analog          =             384
analog_obj      = <softusbduino.pin.AnalogInputValue object at 0x876290c>
base            = <softusbduino.pin.Pins object at 0x8746bcc>
digital         =               0
digital_in      =               0
digital_out     =            None
is_analog       =           False
is_digital      =            True
is_usb_minus    =           False
is_usb_plus     =           False
mcu             = <softusbduino.arduino.Arduino object at 0x868514c>
mode            =               0
name            =              D7
nr              =               7
nr_analog       =            None
programming_function =            None
pwm             = <softusbduino.pwmpin.PwmPin object at 0x876296c>
---------- nr=8 ---------------
A0              =              14
analog          =             386
analog_obj      = <softusbduino.pin.AnalogInputValue object at 0x876272c>
base            = <softusbduino.pin.Pins object at 0x8746bcc>
digital         =               1
digital_in      =               1
digital_out     =            None
is_analog       =           False
is_digital      =            True
is_usb_minus    =           False
is_usb_plus     =           False
mcu             = <softusbduino.arduino.Arduino object at 0x868514c>
mode            =               0
name            =              D8
nr              =               8
nr_analog       =            None
programming_function =            None
pwm             = <softusbduino.pwmpin.PwmPin object at 0x875de4c>
---------- nr=9 ---------------
A0              =              14
analog          =             373
analog_obj      = <softusbduino.pin.AnalogInputValue object at 0x875deac>
base            = <softusbduino.pin.Pins object at 0x8746bcc>
digital         =               1
digital_in      =               1
digital_out     =            None
is_analog       =           False
is_digital      =            True
is_usb_minus    =           False
is_usb_plus     =           False
mcu             = <softusbduino.arduino.Arduino object at 0x868514c>
mode            =               0
name            =              D9
nr              =               9
nr_analog       =            None
programming_function =            None
```

```
pwm              = <softusbduino.pwmpin.PwmPin object at 0x875df0c>
--- pwm ---
available        =            True
base             = <softusbduino.pwmpin.Pwm object at 0x874aacc>
base_divisor     =             512
divisor          =              64
divisors_available = [1, 8, 64, 256, 1024]
frequencies_available = [39062.5, 4882.8125, 610.3515625, 152.587890625, 38.14697265625]
frequency        =     610.3515625
pin              = <softusbduino.pin.Pin object at 0x875ddac>
timer_mode       =               3
timer_register_name =          TCCR1B
---------- nr=10 ---------------
A0               =              14
analog           =             349
analog_obj       = <softusbduino.pin.AnalogInputValue object at 0x875df8c>
base             = <softusbduino.pin.Pins object at 0x8746bcc>
digital          =               0
digital_in       =               0
digital_out      =            None
is_analog        =           False
is_digital       =            True
is_usb_minus     =           False
is_usb_plus      =           False
mcu              = <softusbduino.arduino.Arduino object at 0x868514c>
mode             =               0
name             =             D10
nr               =              10
nr_analog        =            None
programming_function =              SS
pwm              = <softusbduino.pwmpin.PwmPin object at 0x875dfac>
--- pwm ---
available        =            True
base             = <softusbduino.pwmpin.Pwm object at 0x874aacc>
base_divisor     =             512
divisor          =              64
divisors_available = [1, 8, 64, 256, 1024]
frequencies_available = [39062.5, 4882.8125, 610.3515625, 152.587890625, 38.14697265625]
frequency        =     610.3515625
pin              = <softusbduino.pin.Pin object at 0x876290c>
timer_mode       =               3
timer_register_name =          TCCR1B
---------- nr=11 ---------------
A0               =              14
analog           =             378
analog_obj       = <softusbduino.pin.AnalogInputValue object at 0x875dfec>
base             = <softusbduino.pin.Pins object at 0x8746bcc>
digital          =               0
digital_in       =               0
digital_out      =            None
is_analog        =           False
is_digital       =            True
is_usb_minus     =           False
is_usb_plus      =           False
mcu              = <softusbduino.arduino.Arduino object at 0x868514c>
mode             =               0
name             =             D11
nr               =              11
nr_analog        =            None
programming_function =            MOSI
pwm              = <softusbduino.pwmpin.PwmPin object at 0x879756c>
--- pwm ---
available        =            True
```

```
base            = <softusbduino.pwmpin.Pwm object at 0x874aacc>
base_divisor    =             512
divisor         =              64
divisors_available = [1, 8, 32, 64, 128, 256, 1024]
frequencies_available = [39062.5, 4882.8125, 1220.703125, 610.3515625, 305.17578125, 152.587890625
frequency       =     610.3515625
pin             = <softusbduino.pin.Pin object at 0x875deac>
timer_mode      =               4
timer_register_name =          TCCR2B
---------- nr=12 ---------------
A0              =              14
analog          =             373
analog_obj      = <softusbduino.pin.AnalogInputValue object at 0x879764c>
base            = <softusbduino.pin.Pins object at 0x8746bcc>
digital         =               0
digital_in      =               0
digital_out     =            None
is_analog       =           False
is_digital      =            True
is_usb_minus    =           False
is_usb_plus     =           False
mcu             = <softusbduino.arduino.Arduino object at 0x868514c>
mode            =               0
name            =             D12
nr              =              12
nr_analog       =            None
programming_function =            MISO
pwm             = <softusbduino.pwmpin.PwmPin object at 0x87975ec>
---------- nr=13 ---------------
A0              =              14
analog          =             351
analog_obj      = <softusbduino.pin.AnalogInputValue object at 0x879770c>
base            = <softusbduino.pin.Pins object at 0x8746bcc>
digital         =               0
digital_in      =               0
digital_out     =            None
is_analog       =           False
is_digital      =            True
is_usb_minus    =           False
is_usb_plus     =           False
mcu             = <softusbduino.arduino.Arduino object at 0x868514c>
mode            =               0
name            =             D13
nr              =              13
nr_analog       =            None
programming_function =             SCK
pwm             = <softusbduino.pwmpin.PwmPin object at 0x879776c>
---------- nr=14 ---------------
A0              =              14
analog          =             383
analog_obj      = <softusbduino.pin.AnalogInputValue object at 0x87977cc>
base            = <softusbduino.pin.Pins object at 0x8746bcc>
digital         =               0
digital_in      =               0
digital_out     =            None
is_analog       =            True
is_digital      =           False
is_usb_minus    =           False
is_usb_plus     =           False
mcu             = <softusbduino.arduino.Arduino object at 0x868514c>
mode            =               0
name            =              A0
nr              =              14
```

```
nr_analog        =               0
programming_function =            None
pwm              = <softusbduino.pwmpin.PwmPin object at 0x879782c>
---------- nr=15 ---------------
A0               =              14
analog           =             363
analog_obj       = <softusbduino.pin.AnalogInputValue object at 0x879788c>
base             = <softusbduino.pin.Pins object at 0x8746bcc>
digital          =               0
digital_in       =               0
digital_out      =            None
is_analog        =            True
is_digital       =           False
is_usb_minus     =           False
is_usb_plus      =           False
mcu              = <softusbduino.arduino.Arduino object at 0x868514c>
mode             =               0
name             =              A1
nr               =              15
nr_analog        =               1
programming_function =            None
pwm              = <softusbduino.pwmpin.PwmPin object at 0x87978ac>
---------- nr=16 ---------------
A0               =              14
analog           =             350
analog_obj       = <softusbduino.pin.AnalogInputValue object at 0x879794c>
base             = <softusbduino.pin.Pins object at 0x8746bcc>
digital          =               0
digital_in       =               0
digital_out      =            None
is_analog        =            True
is_digital       =           False
is_usb_minus     =           False
is_usb_plus      =           False
mcu              = <softusbduino.arduino.Arduino object at 0x868514c>
mode             =               0
name             =              A2
nr               =              16
nr_analog        =               2
programming_function =            None
pwm              = <softusbduino.pwmpin.PwmPin object at 0x87979ac>
---------- nr=17 ---------------
A0               =              14
analog           =             351
analog_obj       = <softusbduino.pin.AnalogInputValue object at 0x8797a0c>
base             = <softusbduino.pin.Pins object at 0x8746bcc>
digital          =               0
digital_in       =               0
digital_out      =            None
is_analog        =            True
is_digital       =           False
is_usb_minus     =           False
is_usb_plus      =           False
mcu              = <softusbduino.arduino.Arduino object at 0x868514c>
mode             =               0
name             =              A3
nr               =              17
nr_analog        =               3
programming_function =            None
pwm              = <softusbduino.pwmpin.PwmPin object at 0x8797a2c>
---------- nr=18 ---------------
A0               =              14
analog           =             364
```

```
analog_obj      = <softusbduino.pin.AnalogInputValue object at 0x8797acc>
base            = <softusbduino.pin.Pins object at 0x8746bcc>
digital         =               0
digital_in      =               0
digital_out     =            None
is_analog       =            True
is_digital      =           False
is_usb_minus    =           False
is_usb_plus     =           False
mcu             = <softusbduino.arduino.Arduino object at 0x868514c>
mode            =               0
name            =              A4
nr              =              18
nr_analog       =               4
programming_function =            None
pwm             = <softusbduino.pwmpin.PwmPin object at 0x8797b2c>
---------- nr=19 ---------------
A0              =              14
analog          =             341
analog_obj      = <softusbduino.pin.AnalogInputValue object at 0x8797b8c>
base            = <softusbduino.pin.Pins object at 0x8746bcc>
digital         =               0
digital_in      =               0
digital_out     =            None
is_analog       =            True
is_digital      =           False
is_usb_minus    =           False
is_usb_plus     =           False
mcu             = <softusbduino.arduino.Arduino object at 0x868514c>
mode            =               0
name            =              A5
nr              =              19
nr_analog       =               5
programming_function =            None
pwm             = <softusbduino.pwmpin.PwmPin object at 0x8797bec>

================================
Arduino().vcc attributes:
================================
_u_voltage      = 4.87619047619+/-0.042218099361
base            = <softusbduino.arduino.Arduino object at 0x868514c>
t               =   1330892696.64
u_voltage       = 4.87619047619+/-0.042218099361
voltage         =   4.87619047619

================================
Arduino().defines attributes:
================================
base            = <softusbduino.defines.DefinesLowLevel object at 0x87466cc>
intdef_ids      = Bunch(A0=14, ARDUINO=9, E2END=2, E2PAGESIZE=5, FLASHEND=3, F_CPU=8, MISO=12, MO
special_defines = Bunch(MAGIC_NUMBER=42, MCU_DEFINED=26, USBDRV_VERSION=25, USB_CFG_IOPORT=34, __

================================
Arduino().registers attributes:
================================
base            = <softusbduino.registers.RegistersLowLevel object at 0x874aa4c>
register_id_map = Bunch(AC0CON=0, AC1CON=1, AC1ECON=2, AC2CON=3, AC2ECON=4, AC3CON=5, AC3ECON=6, 


================================
defines:
================================
A0                   =                 14
```

```
ARDUINO              =                 22
E2END                =                511
E2PAGESIZE           =                  4
FLASHEND             =               8191
F_CPU                =           20000000
MAGIC_NUMBER         =                 42
MCU_DEFINED          =   __AVR_ATmega88__
MISO                 =                 12
MOSI                 =                 11
ONEWIRE_BUS_COUNT    =           67699460
RAMEND               =               1279
SCK                  =                 13
SOFTUSBDUINO_FIRMWARE_VERSION =                  2
SOFTUSBDUINO_VERSION =              10000
SPM_PAGESIZE         =                 64
SS                   =                 10
USBDRV_VERSION       =           20100715
USB_CFG_DMINUS_BIT   =                  4
USB_CFG_DPLUS_BIT    =                  2
USB_CFG_IOPORT       =                  4
XRAMEND              =               1279
__AVR_LIBC_DATE_     =           20110216
__AVR_LIBC_VERSION__ =              10701
__DATE__             =        Mar  1 2012
__TIME__             =           16:39:54

================================
registers:
================================
ACSR                 = 0x10 @0x50
ADCH                 = 0x01 @0x79
ADCL                 = 0x67 @0x78
ADCSRA               = 0x97 @0x7A
ADCSRB               = 0x00 @0x7B
ADMUX                = 0x45 @0x7C
ASSR                 = 0x00 @0xB6
CLKPR                = 0x00 @0x61
DDRB                 = 0x00 @0x24
DDRC                 = 0x00 @0x27
DDRD                 = 0x00 @0x2A
DIDR0                = 0x00 @0x7E
DIDR1                = 0x00 @0x7F
EEAR                 = 0xFF @0x41
EEARH                = 0x00 @0x42
EEARL                = 0xFF @0x41
EECR                 = 0x00 @0x3F
EEDR                 = 0x00 @0x40
EICRA                = 0x02 @0x69
EIFR                 = 0x00 @0x3C
EIMSK                = 0x01 @0x3D
GPIOR0               = 0x00 @0x3E
GPIOR1               = 0x00 @0x4A
GPIOR2               = 0x00 @0x4B
GTCCR                = 0x00 @0x43
ICR1                 = 0x00 @0x86
ICR1H                = 0x00 @0x87
ICR1L                = 0x00 @0x86
MCUCR                = 0x00 @0x55
MCUSR                = 0x03 @0x54
MONDR                = 0x21 @0x51
OCR0A                = 0x00 @0x47
OCR0B                = 0x00 @0x48
OCR1A                = 0x00 @0x88
```

```
OCR1AH               = 0x00 @0x89
OCR1AL               = 0x00 @0x88
OCR1B                = 0x00 @0x8A
OCR1BH               = 0x00 @0x8B
OCR1BL               = 0x00 @0x8A
OCR2A                = 0x00 @0xB3
OCR2B                = 0x00 @0xB4
OSCCAL               = 0x98 @0x66
PCICR                = 0x00 @0x68
PCIFR                = 0x00 @0x3B
PCMSK0               = 0x00 @0x6B
PCMSK1               = 0x00 @0x6C
PCMSK2               = 0x00 @0x6D
PINB                 = 0x03 @0x23
PINC                 = 0x00 @0x26
PIND                 = 0x19 @0x29
PORTB                = 0x00 @0x25
PORTC                = 0x00 @0x28
PORTD                = 0x00 @0x2B
PRR                  = 0x00 @0x64
SMCR                 = 0x00 @0x53
SP                   = 0xE4 @0x5D
SPCR                 = 0x00 @0x4C
SPDR                 = 0xB9 @0x4E
SPH                  = 0x04 @0x5E
SPL                  = 0xE4 @0x5D
SPMCSR               = 0x00 @0x57
SPSR                 = 0x00 @0x4D
SREG                 = 0x82 @0x5F
TCCR0A               = 0x03 @0x44
TCCR0B               = 0x03 @0x45
TCCR1A               = 0x01 @0x80
TCCR1B               = 0x03 @0x81
TCCR1C               = 0x00 @0x82
TCCR2A               = 0x01 @0xB0
TCCR2B               = 0x04 @0xB1
TCNT0                = 0x45 @0x46
TCNT1                = 0xC1 @0x84
TCNT1H               = 0x00 @0x85
TCNT1L               = 0x29 @0x84
TCNT2                = 0xD2 @0xB2
TIFR0                = 0x07 @0x35
TIFR1                = 0x07 @0x36
TIFR2                = 0x07 @0x37
TIMSK0               = 0x00 @0x6E
TIMSK1               = 0x00 @0x6F
TIMSK2               = 0x00 @0x70
TWAMR                = 0x00 @0xBD
TWAR                 = 0xFE @0xBA
TWBR                 = 0x00 @0xB8
TWCR                 = 0x00 @0xBC
TWDR                 = 0xFF @0xBB
TWSR                 = 0xF8 @0xB9
UBRR0                = 0x00 @0xC4
UBRR0H               = 0x00 @0xC5
UBRR0L               = 0x00 @0xC4
UCSR0A               = 0x20 @0xC0
UCSR0B               = 0x00 @0xC1
UCSR0C               = 0x06 @0xC2
UDR0                 = 0x00 @0xC6
WDTCSR               = 0x0E @0x60
```

CHAPTER

EIGHT

HARDWARE

http://vusb.wikidot.com/hardware

I use Solution B:

“Solution B: Level conversion on D+ and D- Level conversion with Zener diodes.

Instead of reducing the AVR’s power supply, we can limit the output voltage on D+ and D- with Zener diodes. We recommend 3.6 V low power types, those that look like 1N4148 (usually 500 mW or less). Low power types are required because they have less capacitance and thus cause less distortion on the data lines. And 3.6 V is better than 3.3 V because 3.3 V diodes yield only ca. 2.7 V in conjunction with an 1.5 kâ„¦ (or more exactly 10 kâ„¦) pull-up resistor. With 3.3 V diodes, the device may not be detected reliably.

If you use Zener diodes for level conversion, please measure the voltage levels to make sure that the diodes you have chosen match the requirements.

Advantages of the Zener diode approach:

- Low cost.
- Easy to obtain.
- Entire design can be at 5 V.
- AVR can be clocked at high rates.

Disadvantages:

- Not a clean solution, a compromise between all parameters must be found.
- Zener diodes come with a broad range of characteristics, especially at low currents, results may not be reproducible.
- High currents when sending high-level.
- High level is different for signaling and in idle state because signaling uses high currents to drive the diodes while idle state is driven by a 1.5 kâ„¦ pull-up resistor.”

8.1 Pins

USB pins are defined in `pinconfig.h`:

```
#define USB_CFG_IOPORTNAME      D
/* This is the port where the USB bus is connected. When you configure it to
 * "B", the registers PORTB, PINB and DDRB will be used.
 */
#define USB_CFG_DMINUS_BIT      4
/* This is the bit number in USB_CFG_IOPORT where the USB D- line is connected.
 * This may be any bit in the port.
 */
#define USB_CFG_DPLUS_BIT       2
```

```
/* This is the bit number in USB_CFG_IOPORT where the USB D+ line is connected.
 * This may be any bit in the port. Please note that D+ must also be connected
 * to interrupt pin INT0! [You can also use other interrupts, see section
 * "Optional MCU Description" below, or you can connect D- to the interrupt, as
 * it is required if you use the USB_COUNT_SOF feature. If you use D- for the
 * interrupt, the USB interrupt will also be triggered at Start-Of-Frame
 * markers every millisecond.]
 */
```

Pin mapping depends on board. Example:

http://arduino.cc/hu/Hacking/PinMapping

CHAPTER

NINE

BUILD TESTS

9.1 Results

9.1.1 Arduino version 0022

index	board	Basic	OneWire
1	atmega8	ERR	ERR
2	atmega88	ERR	ERR
3	bt	ERR	ERR
4	bt328	ERR	ERR
5	diecimila	ERR	ERR
6	fio	ERR	ERR
7	lilypad	ERR	ERR
8	lilypad328	ERR	ERR
9	mega	ERR	ERR
10	mega2560	ERR	ERR
11	metaboard	ERR	ERR
12	mini	ERR	ERR
13	pro	ERR	ERR
14	pro328	ERR	ERR
15	pro5v	ERR	ERR
16	pro5v328	ERR	ERR
17	uno	ERR	ERR
18	arduino_OrangutanSVP1284	ERR	ERR
19	arduino_amber128	ERR	ERR
20	arduino_android2561	ERR	ERR
21	arduino_android2561_16	ERR	ERR
22	arduino_at90can128	ERR	ERR
23	arduino_at90can32	ERR	ERR
24	arduino_at90can64	ERR	ERR
25	arduino_at90usb162	ERR	ERR
26	arduino_at90usb646	ERR	ERR
27	arduino_at90usb647	ERR	ERR
28	arduino_at90usbkey	ERR	ERR
29	arduino_atmega16	ERR	ERR
30	arduino_atmega165	ERR	ERR
31	arduino_atmega3290p	ERR	ERR
32	arduino_atmega8515	ERR	ERR
33	arduino_atmega8535	ERR	ERR
34	arduino_attiny2313	ERR	ERR
35	arduino_attiny26	ERR	ERR

Continued on next page

Table 9.1 – continued from previous page

index	board	Basic	OneWire
36	arduino_attiny45	ERR	ERR
37	arduino_attiny85	ERR	ERR
38	arduino_bahbots1284p	ERR	ERR
39	arduino_butterfly	ERR	ERR
40	arduino_cerebot_plus	ERR	ERR
41	arduino_cerebotii	ERR	ERR
42	arduino_digilent_explorer	ERR	ERR
43	arduino_duino644	ERR	ERR
44	arduino_duino644p	ERR	ERR
45	arduino_gator	ERR	ERR
46	arduino_illuminato	ERR	ERR
47	arduino_penguino_avr	ERR	ERR
48	arduino_teensy2_ser	ERR	ERR
49	arduino_teensypp2_ser	ERR	ERR
50	arduino_wiring1281	ERR	ERR
51	atmega168	ERR	ERR
52	atmega328	ERR	ERR
53	atmega48	ERR	ERR
54	atmega640	ERR	ERR
55	atmega8	ERR	ERR
56	atmega88	ERR	ERR
57	bt	ERR	ERR
58	bt328	ERR	ERR
59	diecimila	ERR	ERR
60	dvk90can1	ERR	ERR
61	ecavr_atmega32	ERR	ERR
62	fio	ERR	ERR
63	lilypad	ERR	ERR
64	lilypad328	ERR	ERR
65	mega	ERR	ERR
66	mega1280stk500v2	ERR	ERR
67	mega2560stk500v2	ERR	ERR
68	mini	ERR	ERR
69	pro	ERR	ERR
70	pro328	ERR	ERR
71	pro5v	ERR	ERR
72	pro5v328	ERR	ERR
73	stk502	ERR	ERR
74	stk525	ERR	ERR
75	stk525_647	ERR	ERR

9.1.2 Arduino version 0023

index	board	Basic	OneWire
76	atmega168	ERR	ERR
77	atmega328	ERR	ERR
78	atmega8	ERR	ERR
79	bt	ERR	ERR
80	bt328	ERR	ERR
81	diecimila	ERR	ERR
82	fio	ERR	ERR
83	lilypad	ERR	ERR
84	lilypad328	ERR	ERR
85	mega	ERR	ERR
86	mega2560	ERR	ERR
87	mini	ERR	ERR
88	pro	ERR	ERR
89	pro328	ERR	ERR
90	pro5v	ERR	ERR
91	pro5v328	ERR	ERR
92	uno	ERR	ERR

9.1.3 Arduino version 1.0

index	board	Basic	OneWire
93	atmega168	ERR	ERR
94	atmega328	ERR	ERR
95	atmega8	ERR	ERR
96	bt	ERR	ERR
97	bt328	ERR	ERR
98	diecimila	ERR	ERR
99	ethernet	ERR	ERR
100	fio	ERR	ERR
101	lilypad	ERR	ERR
102	lilypad328	ERR	ERR
103	mega	ERR	ERR
104	mega2560	ERR	ERR
105	mini	ERR	ERR
106	mini328	ERR	ERR
107	nano	ERR	ERR
108	nano328	ERR	ERR
109	pro	ERR	ERR
110	pro328	ERR	ERR
111	pro5v	ERR	ERR
112	pro5v328	ERR	ERR
113	uno	ERR	ERR

9.2 Board configurations

9.2.1 Arduino version 0022

index	package	id	name	MCU
1	arduino	atmega8	Arduino NG or older w/ ATmega8	atmega8

Cor

Table 9.2 – continued from previous page

index	package	id	name	MCU
2	arduino	atmega88	atmega88@20000000 programmer:usbasp	atmega8
3	arduino	bt	Arduino BT w/ ATmega168	atmega1
4	arduino	bt328	Arduino BT w/ ATmega328	atmega3
5	arduino	diecimila	Arduino Diecimila, Duemilanove, or Nano w/ ATmega168	atmega1
6	arduino	fio	Arduino Fio	atmega3
7	arduino	lilypad	LilyPad Arduino w/ ATmega168	atmega1
8	arduino	lilypad328	LilyPad Arduino w/ ATmega328	atmega3
9	arduino	mega	Arduino Mega (ATmega1280)	atmega1
10	arduino	mega2560	Arduino Mega 2560	atmega2
11	arduino	metaboard	Metaboard	atmega1
12	arduino	mini	Arduino Mini	atmega1
13	arduino	pro	Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega168	atmega1
14	arduino	pro328	Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328	atmega3
15	arduino	pro5v	Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega168	atmega1
16	arduino	pro5v328	Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328	atmega3
17	arduino	uno	Arduino Uno	atmega3
18	arduino-extras	arduino_OrangutanSVP1284	Arduino-Orangutan SVP-1284	atmega1
19	arduino-extras	arduino_amber128	Arduino-Amber 128 14.7456 Mhz	atmega1
20	arduino-extras	arduino_android2561	Arduino-Android 2561 8Mhz	atmega2
21	arduino-extras	arduino_android2561_16	Arduino-Android 2561 16Mhz	atmega2
22	arduino-extras	arduino_at90can128	AT90CAN128 development board NHL (arduino core)	at90can
23	arduino-extras	arduino_at90can32	at90can32 (arduino core)	at90can3
24	arduino-extras	arduino_at90can64	at90can64 (arduino core)	at90can6
25	arduino-extras	arduino_at90usb162	Arduino-at90usb162	at90usb
26	arduino-extras	arduino_at90usb646	Arduino-at90usb646	at90usb6
27	arduino-extras	arduino_at90usb647	Arduino-at90usb647	at90usb6
28	arduino-extras	arduino_at90usbkey	Arduino-at90usbkey	at90usb
29	arduino-extras	arduino_atmega16	Arduino-Atmega16	atmega1
30	arduino-extras	arduino_atmega165	Arduino-Atmega165	atmega1
31	arduino-extras	arduino_atmega3290p	Arduino-Atmega3290p	atmega3
32	arduino-extras	arduino_atmega8515	Arduino-ATmega8515	atmega8
33	arduino-extras	arduino_atmega8535	Arduino-Test-Atmega8535	atmega8
34	arduino-extras	arduino_attiny2313	Arduino-ATtiny2313	attiny23
35	arduino-extras	arduino_attiny26	Arduino-ATtiny26	attiny26
36	arduino-extras	arduino_attiny45	Arduino-ATtiny45	attiny45
37	arduino-extras	arduino_attiny85	Arduino-ATtiny85	attiny85
38	arduino-extras	arduino_bahbots1284p	Arduino-BahBots 1284p	atmega1
39	arduino-extras	arduino_butterfly	Arduino-Butterfly stk500	atmega1
40	arduino-extras	arduino_cerebot_plus	Arduino-Cerebot Plus	atmega2
41	arduino-extras	arduino_cerebotii	Arduino-Cerebot II atemga64	atmega6
42	arduino-extras	arduino_digilent_explorer	Arduino-Digilent I/O Explorer USB	atmega1
43	arduino-extras	arduino_duino644	Arduino-Duino 644	atmega6
44	arduino-extras	arduino_duino644p	Arduino-Duino 644P	atmega6
45	arduino-extras	arduino_gator	Arduino-Rugged Circuits Gator Board	atmega3
46	arduino-extras	arduino_illuminato	Arduino-illuminato	atmega6
47	arduino-extras	arduino_penguino_avr	Arduino-Penguino AVR	atmega3
48	arduino-extras	arduino_teensy2_ser	Arduino-Teensy 2.0 (USB Serial)	atmega3
49	arduino-extras	arduino_teensypp2_ser	Arduino-Teensy++ 2.0 (USB Serial)	at90usb
50	arduino-extras	arduino_wiring1281	Arduino-Wiring 1281	atmega1
51	arduino-extras	atmega168	Arduino NG or older w/ ATmega168	atmega1
52	arduino-extras	atmega328	Arduino Duemilanove or Nano w/ ATmega328	atmega3
53	arduino-extras	atmega48	Arduino Atmega48	atmega4
54	arduino-extras	atmega640	Arduino atmega640	atmega6

Con

Table 9.2 – continued from previous page

index	package	id	name	MCU
55	arduino-extras	atmega8	Arduino NG or older w/ ATmega8	atmega8
56	arduino-extras	atmega88	Atmega88	atmega8
57	arduino-extras	bt	Arduino BT w/ ATmega168	atmega1
58	arduino-extras	bt328	Arduino BT w/ ATmega328	atmega3
59	arduino-extras	diecimila	Arduino Diecimila, Duemilanove, or Nano w/ ATmega168	atmega1
60	arduino-extras	dvk90can1	STK500 w/DVK90CAN1 - AT90can128 (Arduino Core)	at90can
61	arduino-extras	ecavr_atmega32	Embedded market atmega32	atmega3
62	arduino-extras	fio	Arduino Fio	atmega3
63	arduino-extras	lilypad	LilyPad Arduino w/ ATmega168	atmega1
64	arduino-extras	lilypad328	LilyPad Arduino w/ ATmega328	atmega3
65	arduino-extras	mega	Arduino Mega	atmega1
66	arduino-extras	mega1280stk500v2	Arduino Mega1280 stk500v2	atmega1
67	arduino-extras	mega2560stk500v2	Arduino Mega2560 stk500v2	atmega2
68	arduino-extras	mini	Arduino Mini	atmega1
69	arduino-extras	pro	Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega168	atmega1
70	arduino-extras	pro328	Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328	atmega3
71	arduino-extras	pro5v	Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega168	atmega1
72	arduino-extras	pro5v328	Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328	atmega3
73	arduino-extras	stk502	STK500 w/STKk502 - ATmega169 (Arduino Core)	atmega1
74	arduino-extras	stk525	STK500 w/STK525 - at90usb1287 (Arduino Core)	at90usb
75	arduino-extras	stk525_647	STK500 w/STK525 - at90usb647 (Arduino Core)	at90usb6

9.2.2 Arduino version 0023

index	package	id	name	MCU	F_CPU
76	arduino	atmega168	Arduino NG or older w/ ATmega168	atmega168	16000000L
77	arduino	atmega328	Arduino Duemilanove or Nano w/ ATmega328	atmega328p	16000000L
78	arduino	atmega8	Arduino NG or older w/ ATmega8	atmega8	16000000L
79	arduino	bt	Arduino BT w/ ATmega168	atmega168	16000000L
80	arduino	bt328	Arduino BT w/ ATmega328	atmega328p	16000000L
81	arduino	diecimila	Arduino Diecimila, Duemilanove, or Nano w/ ATmega168	atmega168	16000000L
82	arduino	fio	Arduino Fio	atmega328p	8000000L
83	arduino	lilypad	LilyPad Arduino w/ ATmega168	atmega168	8000000L
84	arduino	lilypad328	LilyPad Arduino w/ ATmega328	atmega328p	8000000L
85	arduino	mega	Arduino Mega (ATmega1280)	atmega1280	16000000L
86	arduino	mega2560	Arduino Mega 2560	atmega2560	16000000L
87	arduino	mini	Arduino Mini	atmega168	16000000L
88	arduino	pro	Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega168	atmega168	8000000L
89	arduino	pro328	Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328	atmega328p	8000000L
90	arduino	pro5v	Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega168	atmega168	16000000L
91	arduino	pro5v328	Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328	atmega328p	16000000L
92	arduino	uno	Arduino Uno	atmega328p	16000000L

9.2.3 Arduino version 1.0

index	package	id	name	MCU	F_CPU
93	arduino	atmega168	Arduino NG or older w/ ATmega168	atmega168	16000000L
94	arduino	atmega328	Arduino Duemilanove w/ ATmega328	atmega328p	16000000L
95	arduino	atmega8	Arduino NG or older w/ ATmega8	atmega8	16000000L
96	arduino	bt	Arduino BT w/ ATmega168	atmega168	16000000L
97	arduino	bt328	Arduino BT w/ ATmega328	atmega328p	16000000L
98	arduino	diecimila	Arduino Diecimila or Duemilanove w/ ATmega168	atmega168	16000000L
99	arduino	ethernet	Arduino Ethernet	atmega328p	16000000L
100	arduino	fio	Arduino Fio	atmega328p	8000000L
101	arduino	lilypad	LilyPad Arduino w/ ATmega168	atmega168	8000000L
102	arduino	lilypad328	LilyPad Arduino w/ ATmega328	atmega328p	8000000L
103	arduino	mega	Arduino Mega (ATmega1280)	atmega1280	16000000L
104	arduino	mega2560	Arduino Mega 2560 or Mega ADK	atmega2560	16000000L
105	arduino	mini	Arduino Mini w/ ATmega168	atmega168	16000000L
106	arduino	mini328	Arduino Mini w/ ATmega328	atmega328p	16000000L
107	arduino	nano	Arduino Nano w/ ATmega168	atmega168	16000000L
108	arduino	nano328	Arduino Nano w/ ATmega328	atmega328p	16000000L
109	arduino	pro	Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega168	atmega168	8000000L
110	arduino	pro328	Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328	atmega328p	8000000L
111	arduino	pro5v	Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega168	atmega168	16000000L
112	arduino	pro5v328	Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328	atmega328p	16000000L
113	arduino	uno	Arduino Uno	atmega328p	16000000L

CHAPTER

TEN

DOXYGEN DOCUMENTATION

Files