Lecture 12

Computers and the Real World
Object Oriented Programming
Debugging

Prof. Steven Ludtke
N410.07, sludtke@bcm.edu

What Can CPUs Do ?

Store numbers (1 & 0)

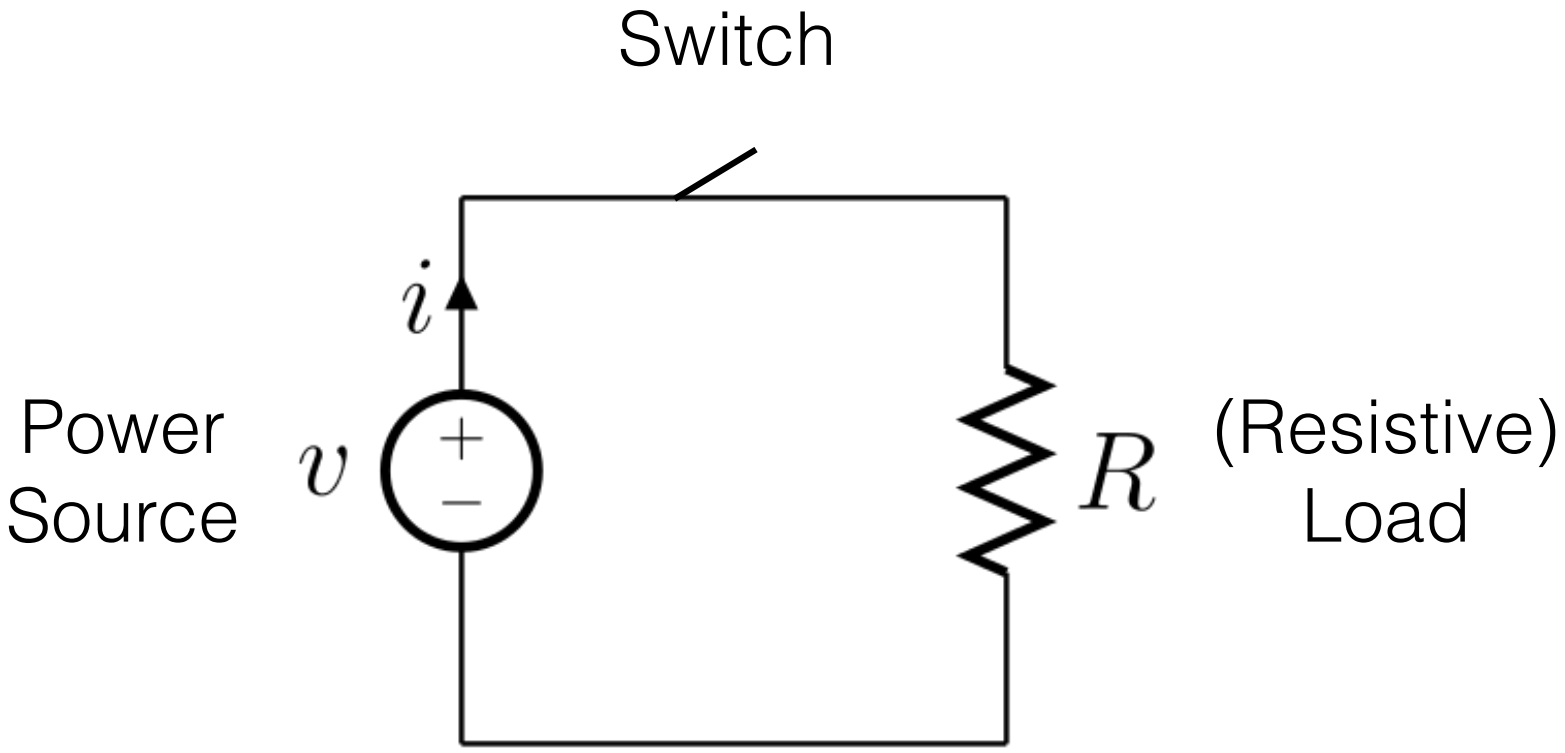
Rearrange stored numbers

Math

Simple decisions based on numbers

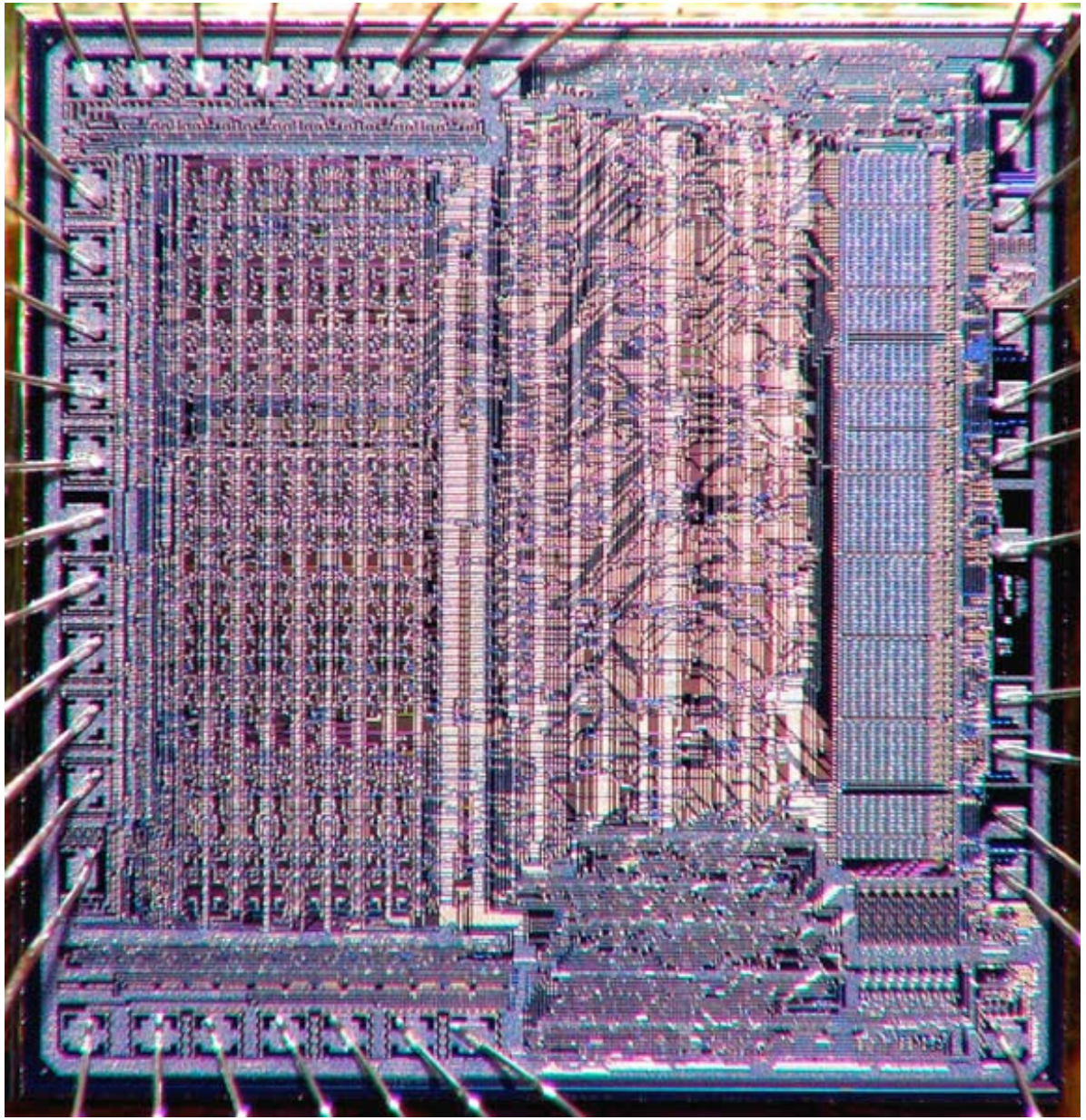
Communicate

Electricity



6800 CPU

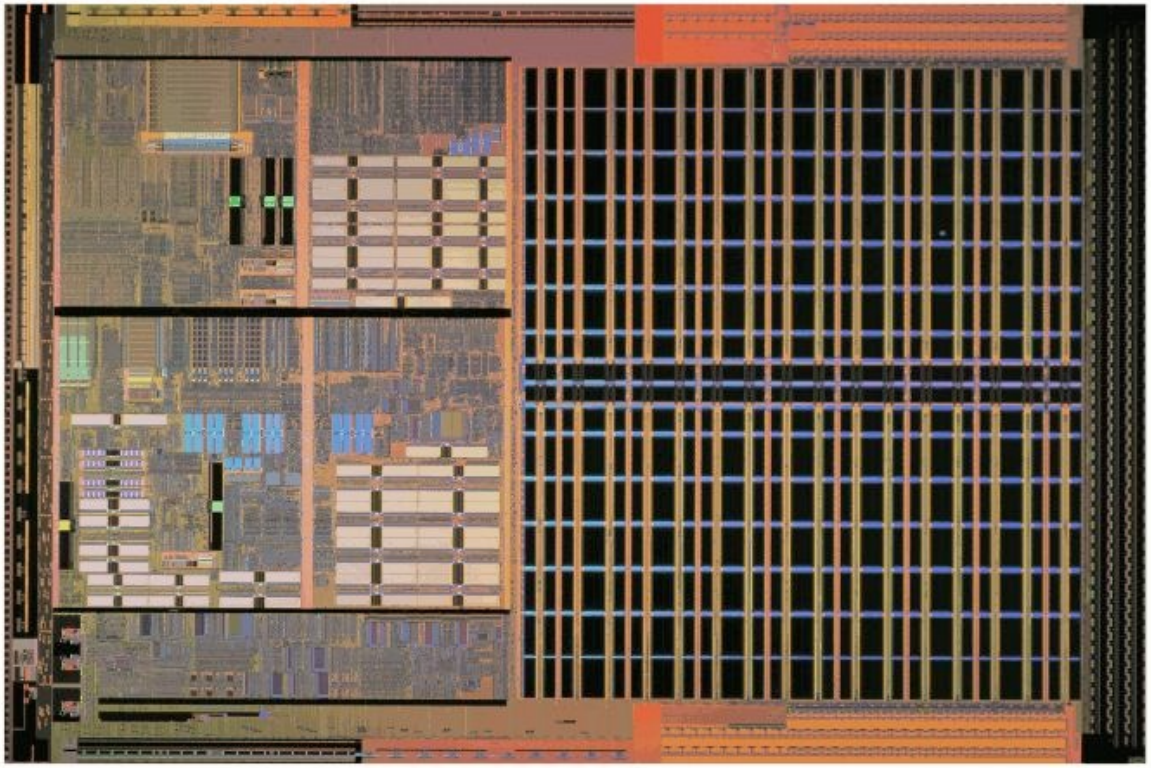
- Introduced 1974
- 4000 transistors
- 1.0-2.5 MHz
- 3, 8 bit registers
- 3, 16 bit registers



www.cpu-world.com

Athlon-64

- ~106 million transistors (~10 m^3 if individually packaged)
- Socket-939 (939 pins)
- 40 bit addressing (1 TB)
- 64 bit data bus
- ~2 GHz
- registers:
- 16, 64 bit integer
- 16, 128 bit 'media'
- 8, 64 bit float



Motorola 6800 CPU

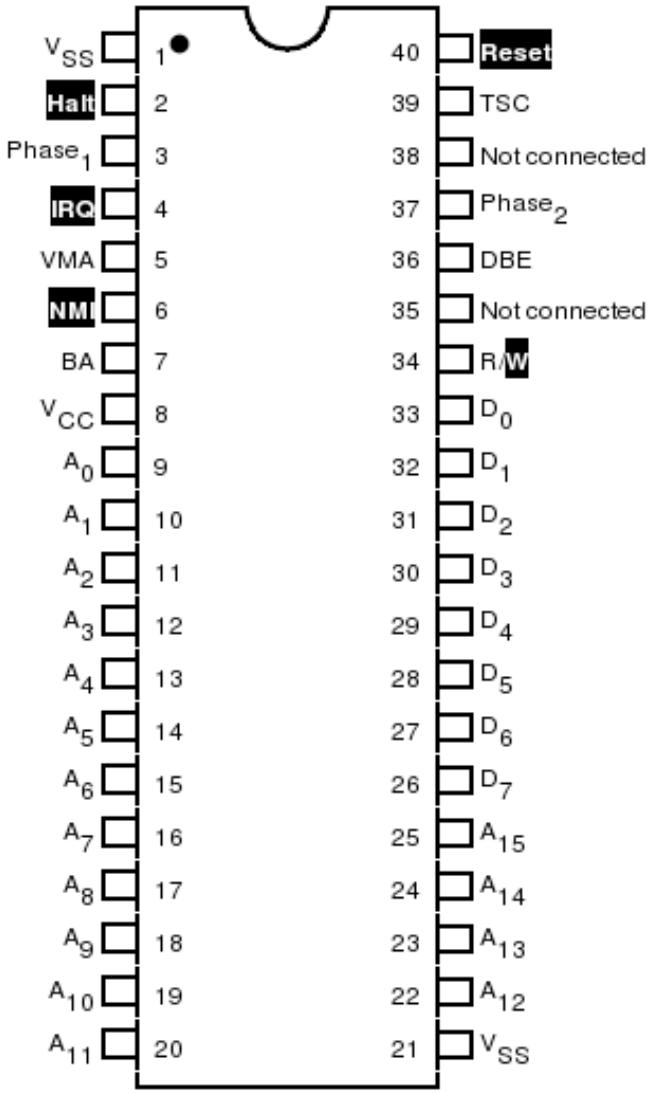
72 instructions (197 opcodes)

8 bit data bus (0-255)

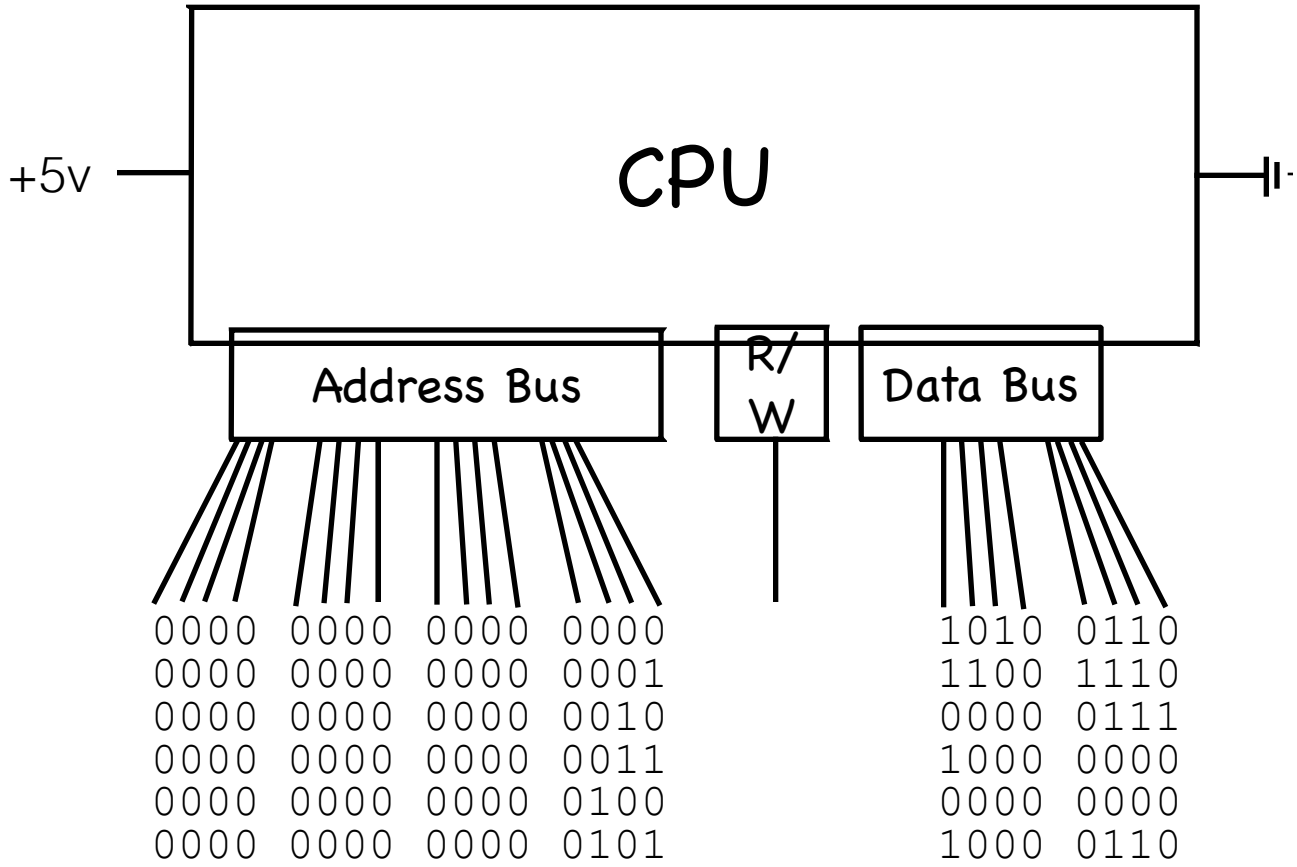
16 bit address bus (64k max RAM)

6 registers:

- 8 bit ACCA
- 8 bit ACCB
- 16 bit IX
- 16 bit PC
- 16 bit SP
- 6 bit CC



Digital Circuits



Communications

- Are computers useful without i/o ?
  - KB/Mouse, video, sound, printer, digital camera, other computers, scanner, TV tuner, home control, specialized equipment (micromill, 3-D printer, telescope control, etc.), scientific instruments (!)
- Memory mapping
- Polling vs Interupts

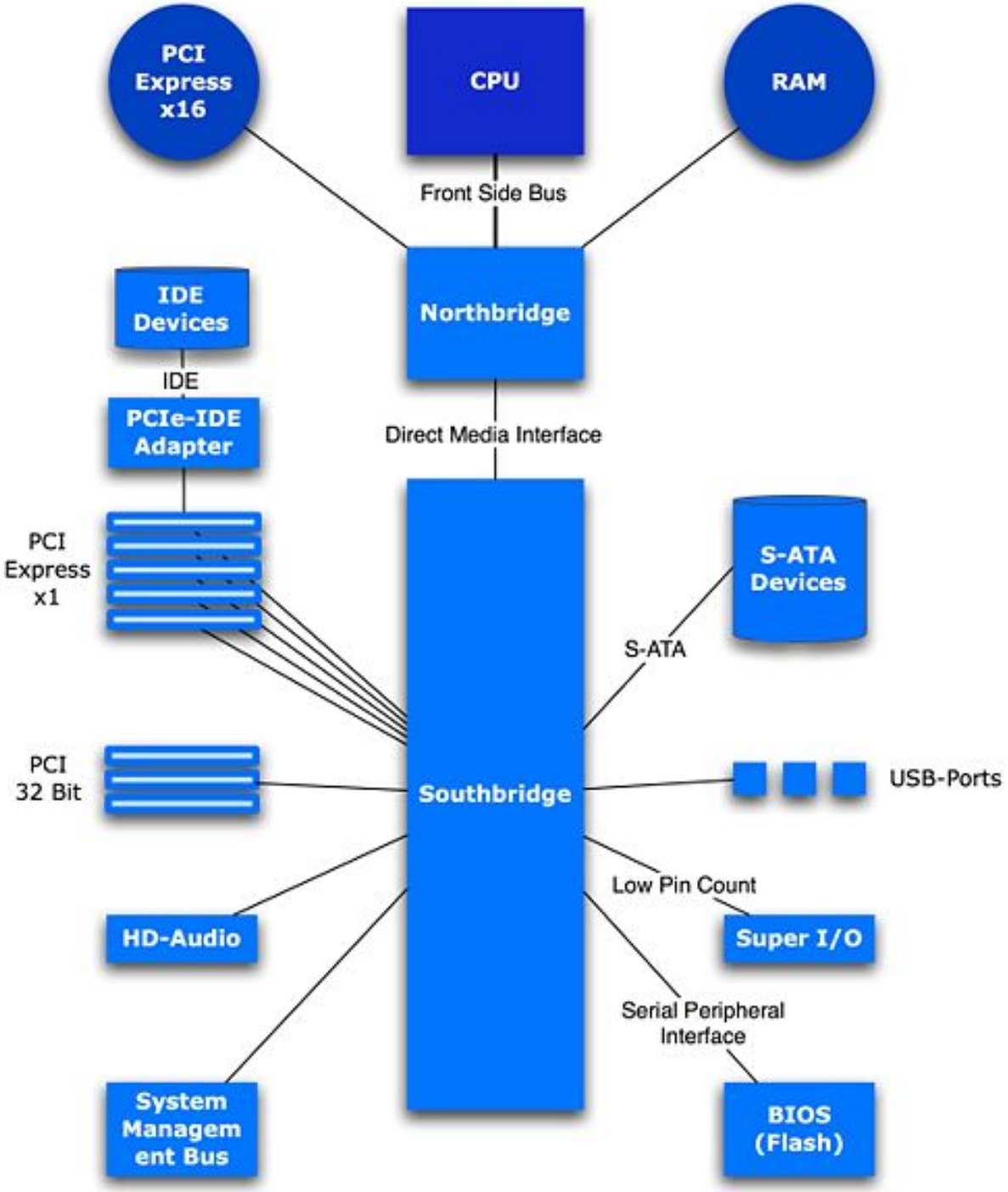
Microprocessors

- 6800 ~4000 transistors, 8 bit (1974)
- 68000 ~70,000 transistors, 16/32 bit (1979)
- 68040 ~1,200,000 transistors 32 bit+FPU (1990)
- Core2 duo, 2 core ~291,000,000 transistors 64 bit (2006)
- Xeon E7v2, 15 core ~4,300,000,000 (2014)
- IPhone6 (A8) ~2,000,000,000 transistors (2014)

Microprocessors

• 1977	$666.66	Apple I	0.0004 GIPS
• 1984	$15,000,000	Cray X-MP/48	0.8 GFLOPS
• 2014	$650	iPhone6	~7.5 GFLOPS(CPU)+
			115 GFLOPS(GPU)

+ 2 cameras, light sensor, accelerometer, magnetic field sensor, GPS microphone, two speakers, 4 radios (cell, wifi, bluetooth,GPS)



http://commons.wikimedia.org/wiki/Image:Typical_intel_chipset.jpg

Software Controlling Hardware ?

	Speed (Mb/s)	Range (m)	description	connector
PCIe (1x/16x)	250/4000	internal	16x mainly for video	
PCI (32/64)	133	internal	64 not widely used	
AGP (1x-8x)	266-2133*	internal	largely obsolete, for video	
network (10/100/1000)	1/12/120	100	1000=gigabit, 10G exists	
802.11 (a/b/g/n)	6/1/6/30	120/250	N is the new standard, most still use G	wireless
bluetooth	0.3	~10	short range device comm	wireless
IRDA	.01-1(?)	1	bluetooth instead	wireless
firewire/1394 (400/800)	50/100	4.5/?	external disks, video	
USB (1/2/3)	1.5/60/500	5/5/3	external disks, etc.	
Thunderbolt	2000	3	disks, monitors, etc.	
Parallel	1	15	pre-USB for printers	
Serial	0.01		old-style modems, specialty devices	
PS/2	n/a		keyboard & mouse	

Disk Interfaces

	Speed (Mb/s)	connector
SATA/SAS	300,600,1200	
SAS		
IDE/ATA	3-133	
Fibre Channel	100-400	optical, long range (>50km)
SCSI	5-640	many connectors, external, short range

Video Interfaces

	HDCP	Description	Connector
HDMI	X	High definition (digital) consumer video	
DVI	X	Newer style computer video, digital, supports HDCP,	
VGA		Old style computer video, supports high resolution, but analog	
Component		Further improved quality, supports higher resolutions	
S-video		Improved quality, still 640x480	
Composite		Oldest style consumer video, poor quality, poor resolution (640x480)	

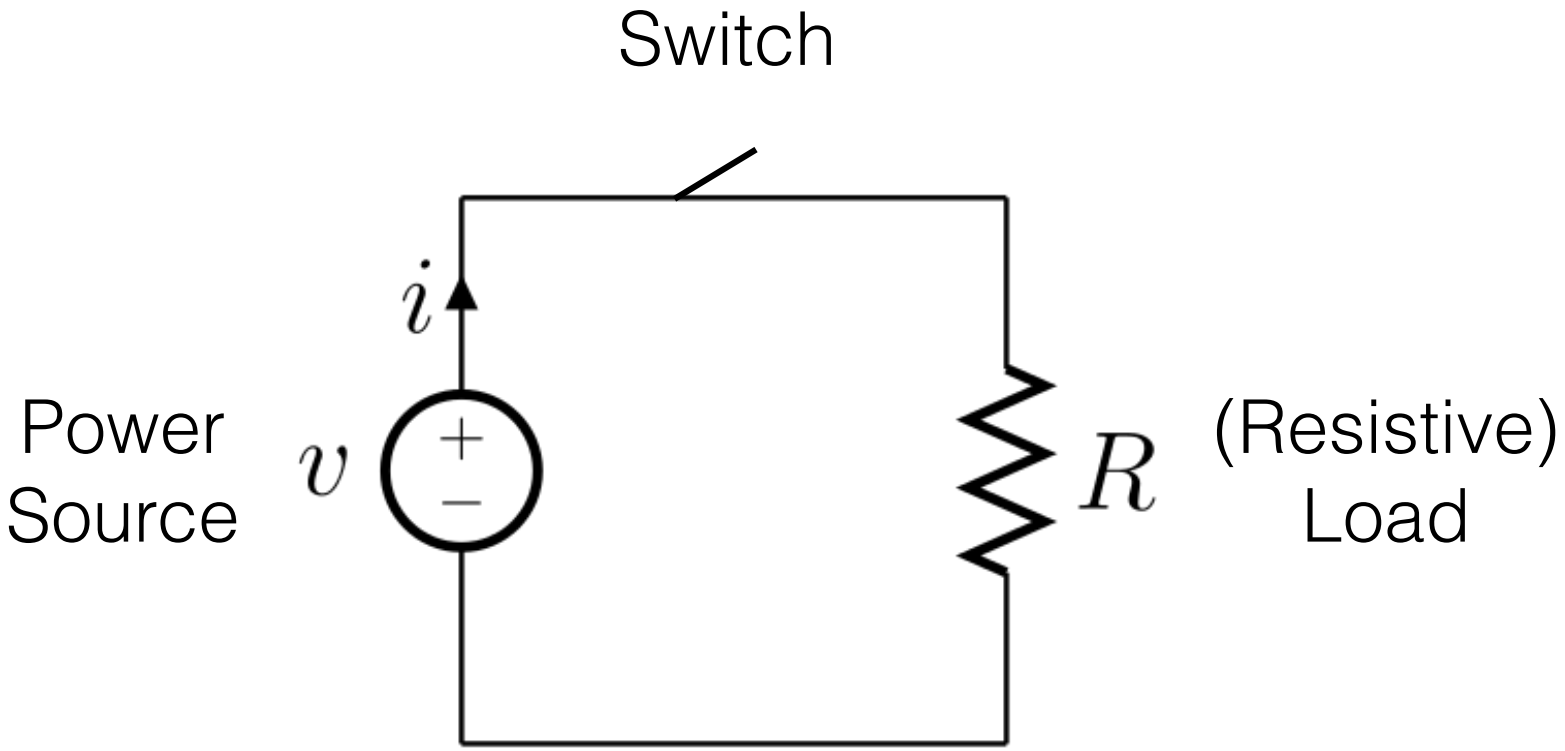
Making

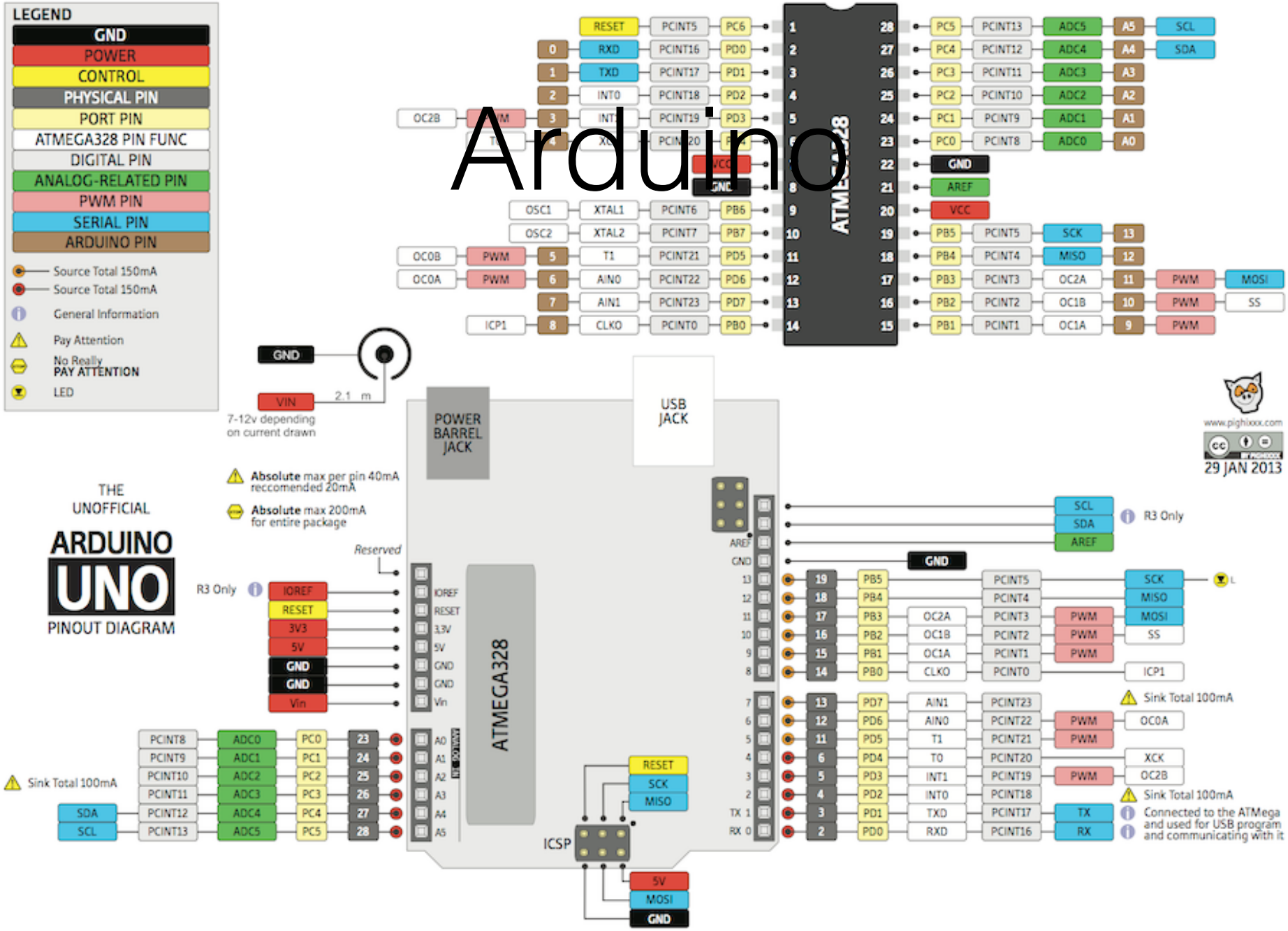
- Computer controlled Mills and Lathes
- 3D Printer
- Embedded processors
  - sensors
  - motors
  - displays

Embedded Computers

		OS	Speed	I/O	Cost
Arduino		None	8/16 Mhz	14-54 digital, 6-14 PWM, 6-16 analog + Shields !	$6-65
Trinket Pro		None	16 Mhz	18 digital, 2 analog, 6 PWM, 1 SPI	$9
BeagleBone Black		Linux + Ard I/O	1 Ghz	HDMI, Ethernet, 2 USB, 65 digital, 8 PWM, 4 timers, 7 analog, 4 serial, 2 I2C, 2 SPI	$45
Raspberry Pi		Linux	700 Mhz +	3 USB, HDMI, Vid, Audio*, Ethernet, SD, Camera, 8 GPIO, I2C, SPI	$35
Intel Edison		Linux	Dual Core 500Mhz	WiFi, Bluetooth 4, I2C, SPI,14 digital, 4 PWM	$40

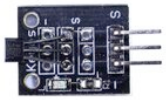
Electricity




LEGEND
GND
POWER
CONTROL
PHYSICAL PIN
PORT PIN
ATMEGA328 PIN FUNC
DIGITAL PIN
ANALOG-RELATED PIN
PWM PIN
SERIAL PIN
ARDUINO PIN
Source Total 150mA
Source Total 150mA
General Information
Pay Attention
No Really
PAY ATTENTION
LED
Arduino
ATMEGA328
GND
VIN
2.1 m
7-12v depending
on current drawn
POWER
BARREL
JACK
USB
JACK
Absolute max per pin 40mA
reccomended 20mA
Absolute max 200mA
for entire package
THE
UNOFFICIAL
ARDUINO
UNO
PINOUT DIAGRAM
Reserved
R3 Only
IOREF
RESET
3V3
5V
GND
GND
Vin
ATMEGA328
ICSP
RESET
SCK
MISO
5V
MOSI
GND
SCL
SDA
AREF
GND
Sink Total 100mA
Sink Total 100mA
Connected to the ATMega
and used for USB program
and communicating with it
www.pighixxx.com
29 JAN 2013

Sensors


Finger-Pulse sensor Module

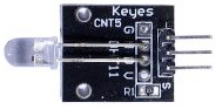

Hall sensor Module


Microphone sensor Module

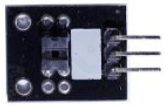

Obstacle avoidance sensor Module


Passive Buzzer Module


Flame sensor Module


Colorful Auto-flash Module


infrared-transmit Module


Light break sensor Module


Magnetic spring Module

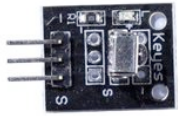

Digital-Temperature sensor Module


High-sensitive voice sensor Module


Photo resistor sensor Module

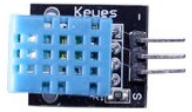

Metal touch sensor Module


hydrargyrum-switch sensor Module


Infrared-receive sensor Module

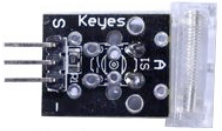

RGB LED Module


Push button Module


humiture sensor Module


Laser-transmit Module


two-color commoncathode LED Module


Knock sensor module


Tracking sensor Module


tilt-switch Module


Shock-switch sensor Module


Rotate-encode Module

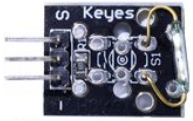

Relay Module

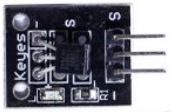

Joystick PS2 Module

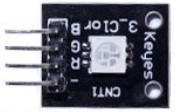

common-cathode RED&GREEN LED Module


Linear-Hall Sensor Module


Active buzzer Module


Magnet-ring sensor Module


18b20 temperature sensor Module


RGB LED Module

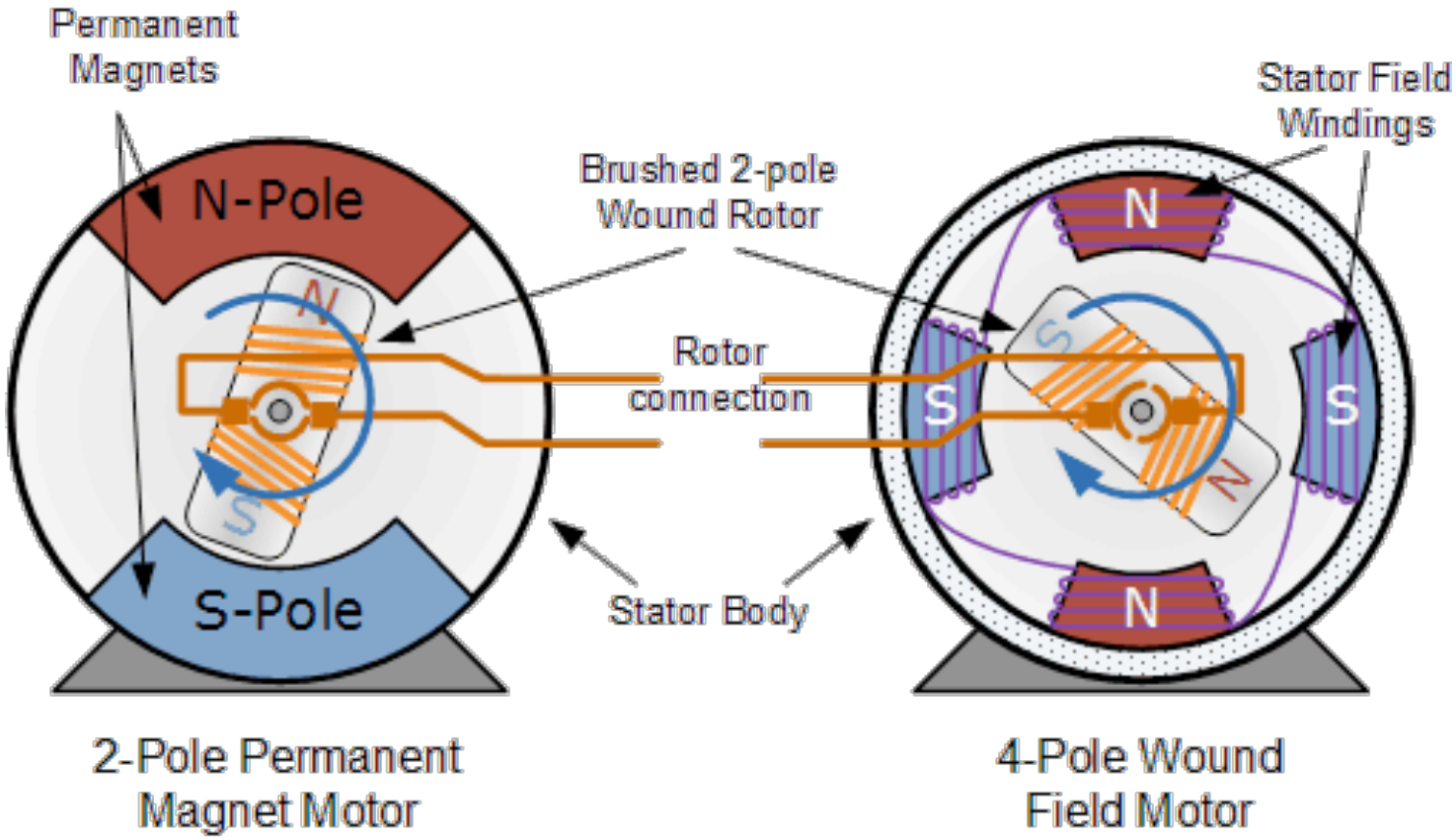

Analog-temperature sensor Module


Magic-ring Module

Motors

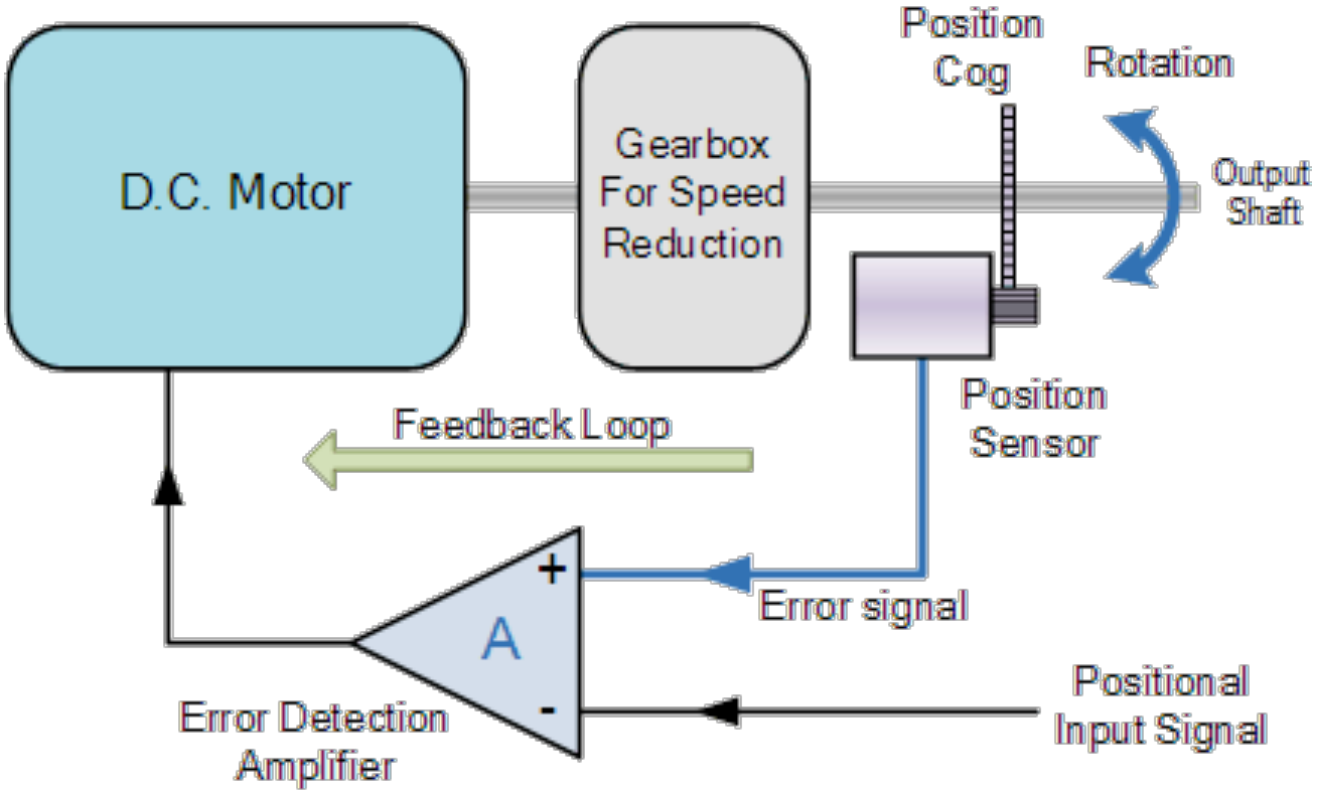
DC/AC Motor	Speed Control	Constant Voltage with Duty Cycle	1 PWM & Power Amp
Gearbox Motor	Speed Control	Same, but slower	1 PWM & Power Amp
Continuous Servo	Digital Speed Control	PWM Controls Speed	1PWM
Standard Servo	Control Angle	PWM Controls Angle	1PWM
Stepper	Digital Positioning	Stepping Sequence	Unipolar: 4 digital + Amp

Standard Motor

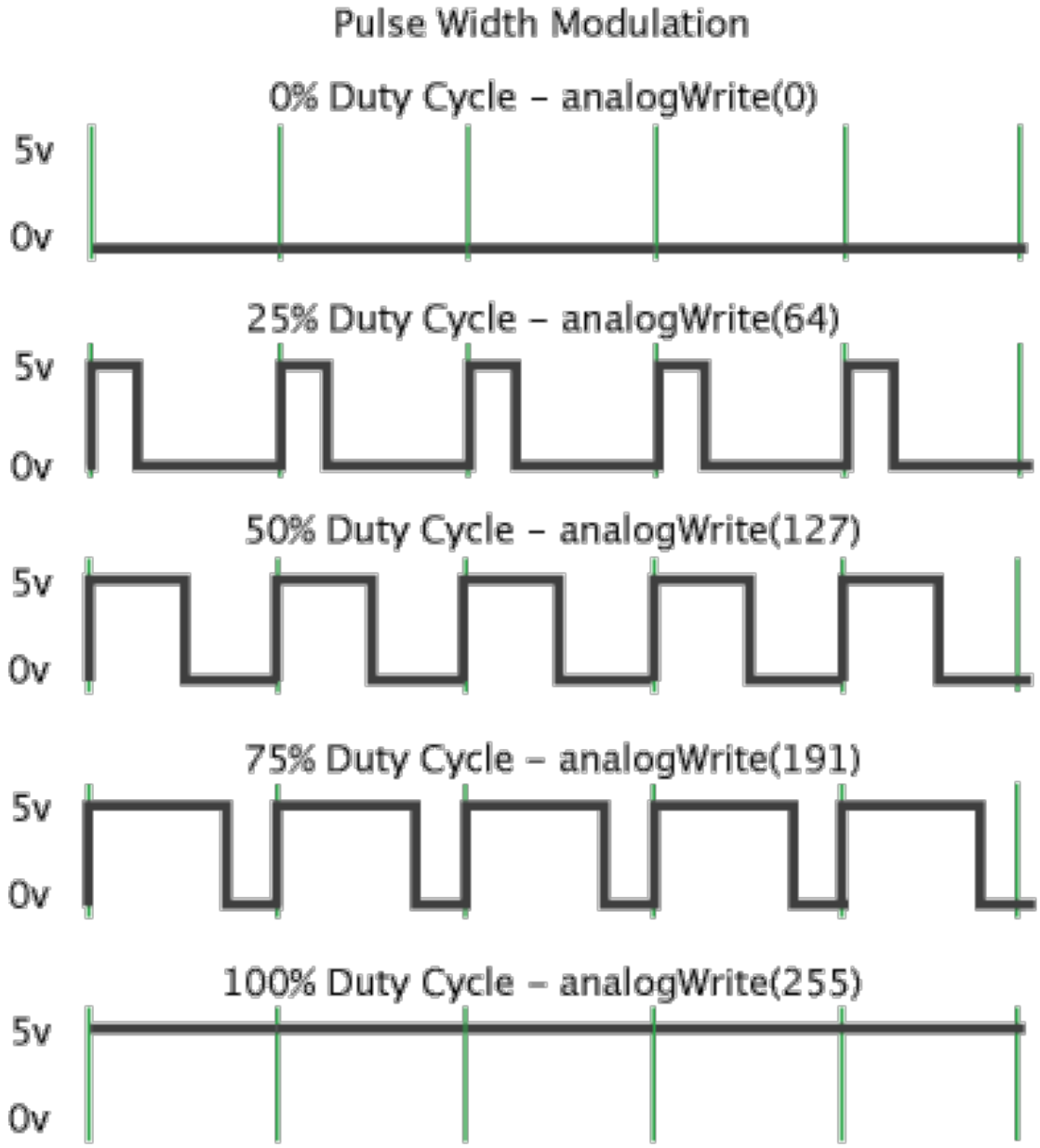


http://www.electronics-tutorials.ws/io/io_7.html

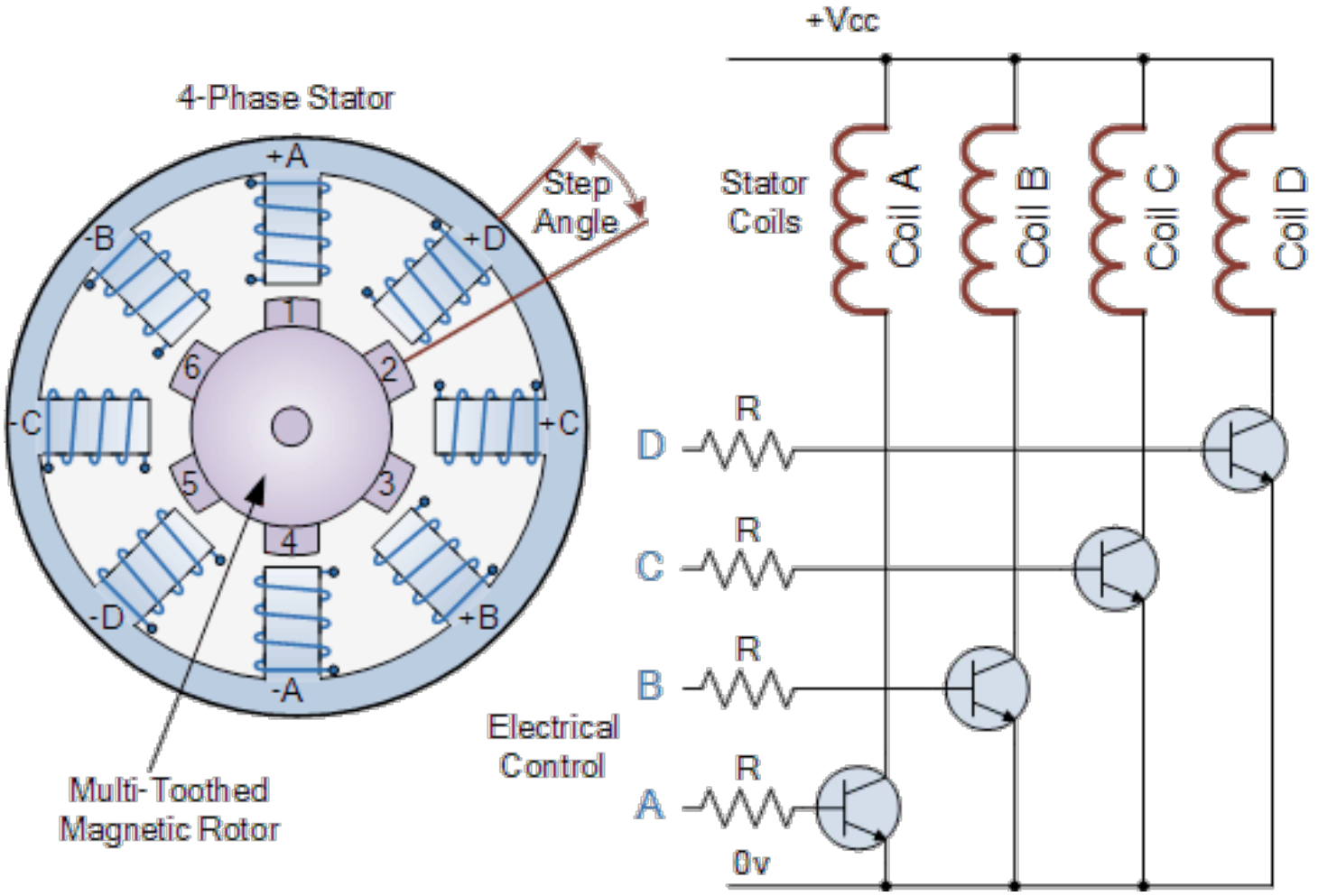
Servo Motor

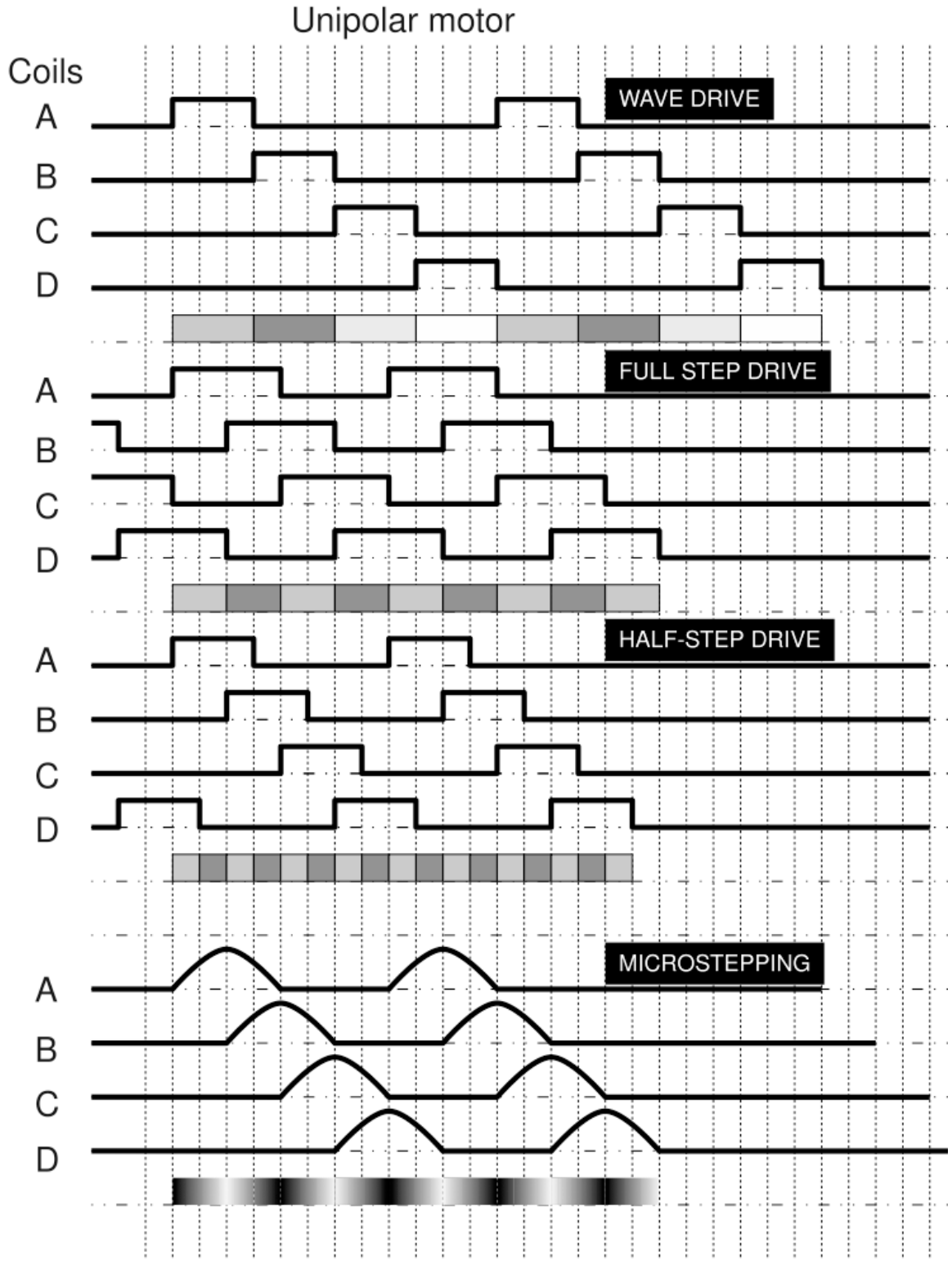

D.C. Motor
Gearbox For Speed Reduction
Position Cog
Rotation
Output Shaft
Position Sensor
Feedback Loop
Error signal
A
+
-
Error Detection Amplifier
Positional Input Signal

PWM


Pulse Width Modulation
0% Duty Cycle – analogWrite(0)
5v
0v
25% Duty Cycle – analogWrite(64)
5v
0v
50% Duty Cycle – analogWrite(127)
5v
0v
75% Duty Cycle – analogWrite(191)
5v
0v
100% Duty Cycle – analogWrite(255)
5v
0v

Stepper Motor




Unipolar motor
Coils
WAVE DRIVE
A
B
C
D
FULL STEP DRIVE
A
B
C
D
HALF-STEP DRIVE
A
B
C
D
MICROSTEPPING
A
B
C
D

How to Store Complex Data ?

- Students
  - Name, address, ...
- Classes
  - Description, Instructor, when offered, ...
- Class Year
  - Which class, year offered, students
- Grades
  - Class, student, grade

XML Example

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<CATALOG>
   <PLANT>
      <COMMON>Bloodroot</COMMON>
      <BOTANICAL>Sanguinaria canadensis</BOTANICAL>
      <ZONE>4</ZONE>
      <LIGHT>Mostly Shady</LIGHT>
      <PRICE CURRENCY=”dollar”>2.44</PRICE>
      <AVAILABILITY>031599</AVAILABILITY>
   </PLANT>
   <PLANT>
      <COMMON>Columbine</COMMON>
      <BOTANICAL>Aquilegia canadensis</BOTANICAL>
      <ZONE>3</ZONE>
      <LIGHT>Mostly Shady</LIGHT>
      <PRICE CURRENCY=”dollar” >9.37</PRICE>
      <AVAILABILITY>030699</AVAILABILITY>
   </PLANT>
</CATALOG>
```

Formatted Files

Genbank

```
  1 gatcctccat atacaacggt atctccacct caggtttaga tctcaacaac ggaaccattg
 61 ccgacatgag acagttaggt atcgtcgaga gttacaagct aaaacgagca gtagtcagct
121 ctgcatctga agccgctgaa gttctactaa gggtggataa catcatccgt gcaagaccaa
181 gaaccgccaa tagacaacat atgtaacata tttaggatat acctcgaaaa taataaaccg
241 ccacactgtc attattataa ttagaaacag aacgcaaaaa ttatccacta tataattcaa
301 agacgcgaaa aaaaaagaac aacgcgtcat agaacttttg gcaattcgcg tcacaaataa
361 attttggcaa cttatgtttc ctcttcgagc agtactcgag ccctgtctca agaatgtaat
421 aatacccatc gtaggtatgg ttaaagatag catctccaca acctcaaagc tccttgccga
481 gagtcgccct cctttgtcga gtaattttca cttttcatat gagaacttat tttcttattc
541 tttactctca catcctgtag tgattgacac tgcaacagcc accatcacta gaagaacaga
601 acaattactt aatagaaaaa ttatatcttc ctcgaaacga tttcctgctt ccaacatcta
661 cgtatatcaa gaagcattca cttaccatga cacagcttca gatttcatta ttgctgacag
```

Formatted Files

PDB

```
HETATM    1  C    FOR A   1A      -3.690  -1.575  -2.801  1.00  0.00           C
HETATM    2  O    FOR A   1A      -3.774  -1.363  -1.586  1.00  0.00           O
HETATM    3  H    FOR A   1A      -4.305  -1.047  -3.545  1.00  0.00           H
ATOM      4  N    VAL A   1       -3.043  -2.601  -3.416  1.00  0.00           N
ATOM      5  CA   VAL A   1       -2.415  -3.640  -2.601  1.00  0.00           C
ATOM      6  C    VAL A   1       -0.920  -3.596  -2.797  1.00  0.00           C
ATOM      7  O    VAL A   1       -0.385  -3.961  -3.851  1.00  0.00           O
ATOM      8  CB   VAL A   1       -2.919  -5.045  -2.884  1.00  0.00           C
ATOM      9  CG1  VAL A   1       -2.163  -6.141  -2.056  1.00  0.00           C
ATOM     10  CG2  VAL A   1       -4.427  -5.132  -2.644  1.00  0.00           C
ATOM     11  H    VAL A   1       -2.760  -2.649  -4.381  1.00  0.00           H
ATOM     12  HA   VAL A   1       -2.651  -3.401  -1.570  1.00  0.00           H
ATOM     13  HB   VAL A   1       -2.542  -5.494  -3.836  1.00  0.00           H
ATOM     14 1HG1  VAL A   1       -2.280  -5.946  -0.974  1.00  0.00           H
ATOM     15 2HG1  VAL A   1       -2.578  -7.146  -2.278  1.00  0.00           H
ATOM     16 3HG1  VAL A   1       -1.080  -6.174  -2.292  1.00  0.00           H
                                      ...
```

pickle

- 'Serialization' - converting a complex object to a stream of data

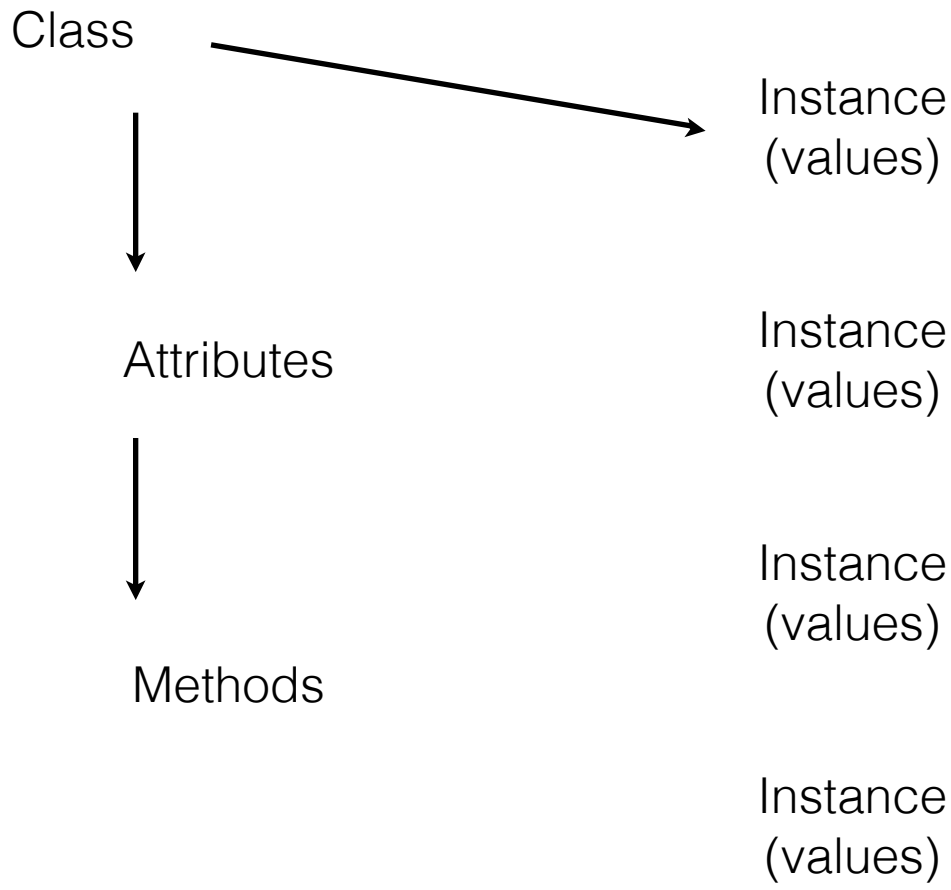
```
from cPickle import dump,load,dumps,loads

dump(obj,file[,protocol])   # stores ‘obj’ in file

obj=load(file)              # restores ‘obj’ from file

str=dumps(obj[,protocol])   # pickled representation of obj

obj=loads(str)              # restore representation of obj
```

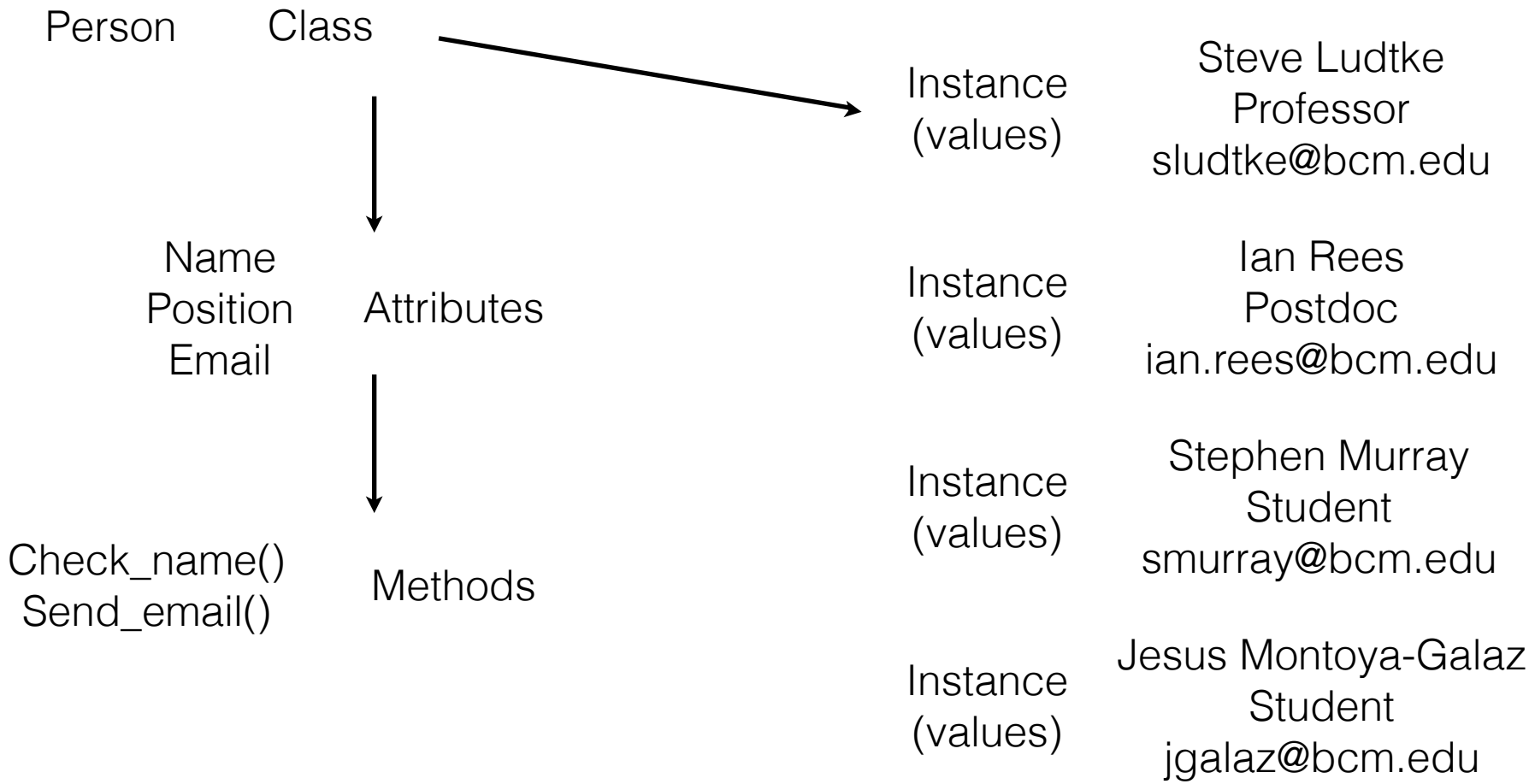
shelve

```
import shelve      # dictionary-like object on disk
dct=shelve.open(filename[,protocol])
dct=shelve.open(filename,writeback=True)
dct.close()
```

Object Oriented Programming (A Quick Introduction)

- A Class is a grouping of associated data elements and optionally code elements (methods).
- class person:
- “This class describes a person”


Class
Instance
(values)
Attributes
Instance
(values)
Instance
(values)
Methods
Instance
(values)


Person
Class
Name
Position
Email
Attributes
Check_name()
Send_email()
Methods
Instance
(values)
Steve Ludtke
Professor
sludtke@bcm.edu
Instance
(values)
Ian Rees
Postdoc
ian.rees@bcm.edu
Instance
(values)
Stephen Murray
Student
smurray@bcm.edu
Instance
(values)
Jesus Montoya-Galaz
Student
jgalaz@bcm.edu

Just Like a Dictionary

```
a={}

a[“firstname”]="Steve"
a[“lastname”]="Ludtke"
a[“address”]="1 Baylor Plaza"

print "%s %s\n%s"%
(a[“firstname”],a[“lastname”],a[“address”])
```

Attributes

```
class Person:
    “This class represents a person”

a=Person()

a.firstname="Steve"
a.lastname="Ludtke"
a.address="1 Baylor Plaza"

print "%s %s\n%s"%(a.firstname,a.lastname,a.address)
```

In C++/Java

```
class person {
   string firstname;
   string lastname;
   string address;
   string city;
   char state[2];
   int zipcode;
   int phone;
};
```

Methods

```
class Person:
    def __init__(self,firstname=None,lastname=None,address=None,
city=None, state=None, zipcode=None, phone=None):
        self.firstname=firstname
        self.lastname=lastname
        self.address=address
        self.city=city
        self.state=state
        self.zipcode=zipcode
        self.phone=phone

    def __repr__(self):
        return "{}, {}\n{}\n{}, {}  {}\n{}\n".format(self.lastname,
self.firstname,self.address,self.city,self.state,self.zipcode,self.phone)

    def fixcase(self):
        self.firstname=self.firstname.title()
        self.lastname=self.lastname.title()
```

Setters & Getters

```
class Person:
…
        def set_name(self,first,last):
                self.firstname=first
                self.lastname=last
                self.fixcase()

        def get_name(self): return "{},
{}".format(self.lastname,self.firstname)

        def get_lastname(self): return self.lastname

        def get_firstname(self): return self.firstname
```

Inheritance

```
class BCM_Person(Person):
    def set_bcmid(self,bcmid):
        self.bcmid=int(bcmid)

    def get_bcmid(self): return self.bcmid
```

Things we skipped

- Scope rules for classes
- Class attributes and methods
- Changes to classes vs instances
- Special methods to emulate, lists, dicts, etc.

Debugging

- print statements ?
- traceback module: print_exc()

Debuggers

- Built in IDLE
- http://wiki.python.org/moin/PythonDebuggers
- http://www-pcmdi.llnl.gov/svn/repository/cdat/trunk/Packages/pydebug/Lib/pydebug.py

Profiling

- 3 built-in profilers
  - profile, cProfile, hotshot
  - python -mcProfile script.py
  - line_profiler
- ‘C’ level profilers
  - valgrind/cachegrind (linux)
  - dtrace (mac)

Lab 5 Preparation

Install (instructions under "the fast way":

- https://learn.adafruit.com/introducing-pro-trinket/setting-up-arduino-ide

Also this plugin:

- https://github.com/adafruit/Adafruit_NeoPixel

- Note: Mac users, you will need to install Java to make this work if you don't already have it installed. There was a Java security flaw about a year ago, and Apple made Java installation optional at that time. The default install should not have any security problems now.