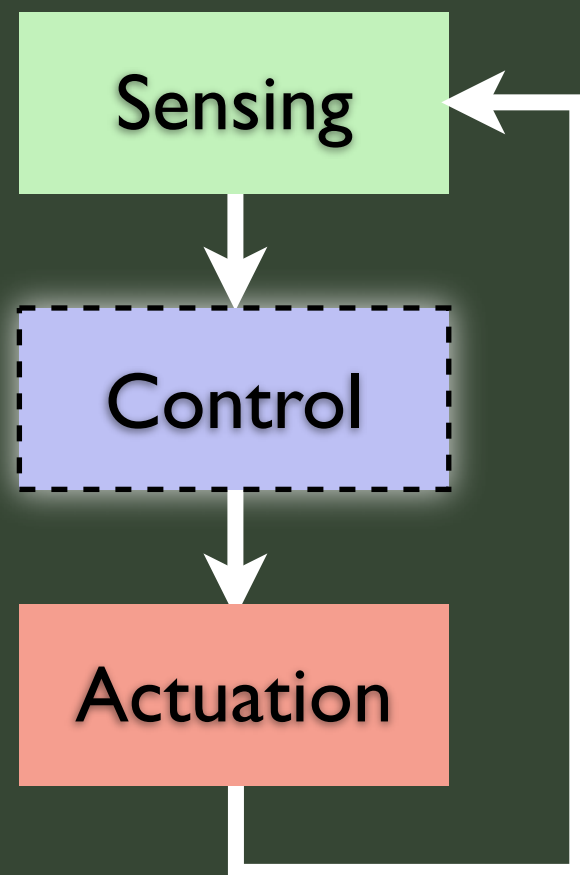


Taming Your Robot with Python

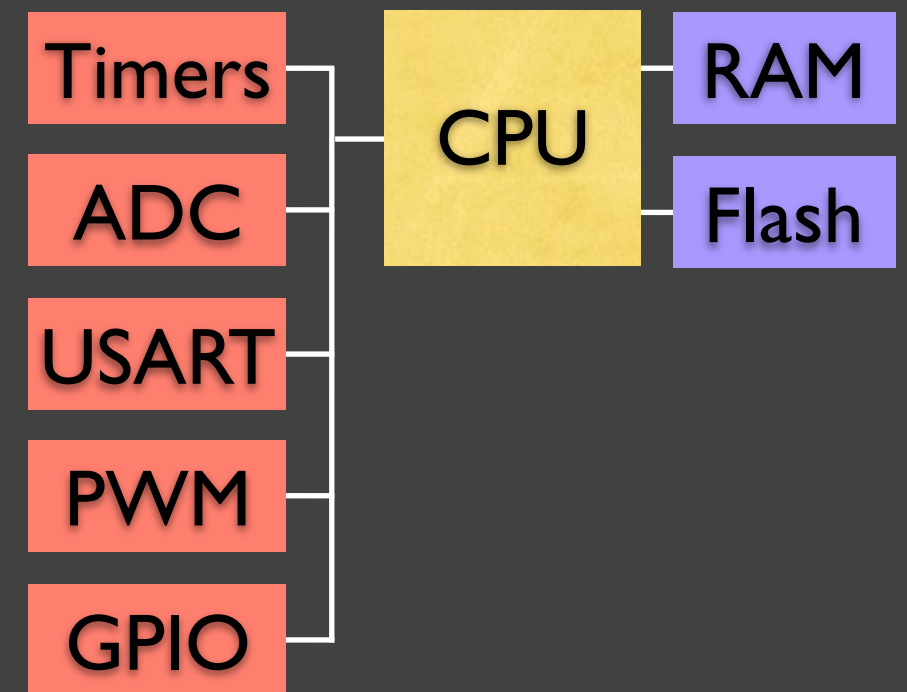
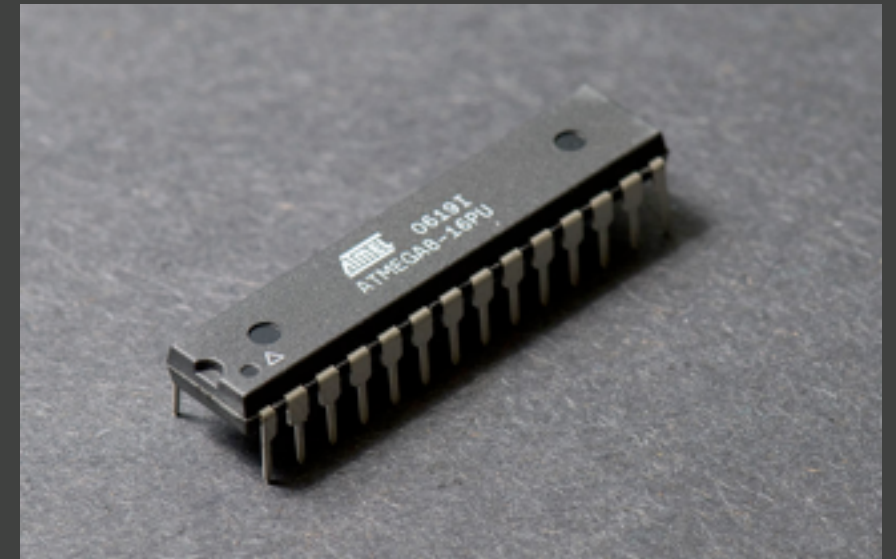
James Snyder

Concepts



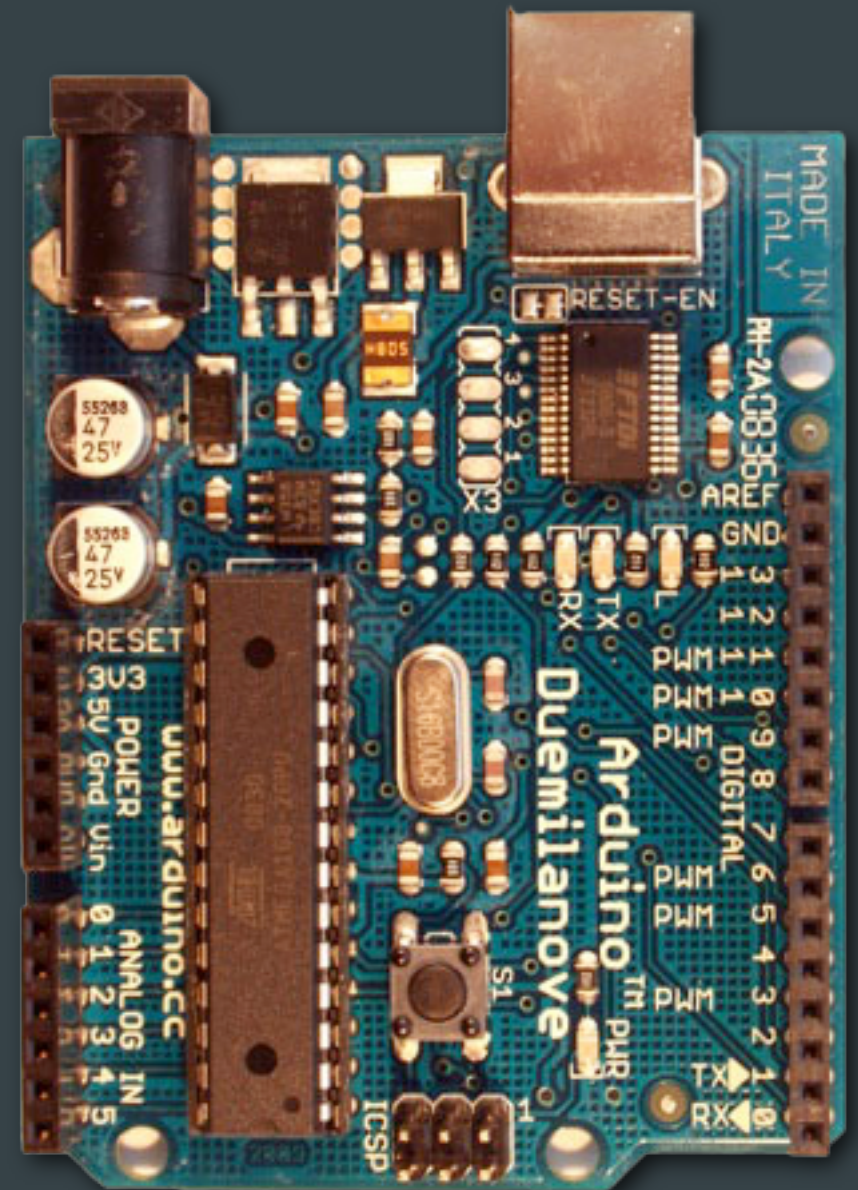
Microcontrollers

- Processor, flash, RAM and peripherals all on one chip
- Inexpensive
- Found in a wide variety of electronics
- Programmable using a wide range of languages



Arduino

- Open source and open design
- Great for prototyping
- GUI IDE based on Processing
- C/C++ & Libraries
- Inexpensive (~\$30)
- Large community, good code examples



Where does Python Come In?

Standard Approach

- Code, compile, flash, test (rinse, repeat)

More Dynamic Approach

- Flash firmware that provides flexible protocol for controlling peripherals
- Collect data and prototype control in Python using pyduino (<http://code.google.com/p/pyduino/>)

Firmata

- Implementation of MIDI message format
- Libraries for various languages, including Python
- Read/write GPIO pins (High/Low, PWM)
- Read analog inputs
- Send/receive messages of arbitrary length

Example Firmata Usage on Arduino

```
#include <Firmata.h>

byte analogPin;

void analogWriteCallback(byte pin, int value)
{
    pinMode(pin,OUTPUT);
    analogWrite(pin, value);
}

void setup()
{
    Firmata.setFirmwareVersion(0, 1);
    Firmata.attach(ANALOG_MESSAGE, analogWriteCallback);
    Firmata.begin();
}

void loop()
{
    while(Firmata.available()) {
        Firmata.processInput();
    }
    for(analogPin = 0; analogPin < TOTAL_ANALOG_PINS; analogPin++) {
        Firmata.sendAnalog(analogPin, analogRead(analogPin));
    }
}
```

Input: Read Voltage From ADC Pin

Arduino

```
(load AnalogFirmata.pde onto Arduino)
```

+

Python

```
import pyduino
ard = pyduino.Arduino('/dev/ttyS0')

pin = 0

# Activate Appropriate Port
ard.analog[pin].set_active(1)

# Read and print ADC values
while 1:
    ard.iterate()
    value = ard.analog[pin].read()
    print "Voltage: %f" % (value*5)
```



Demo: printanalog.py

Output: Pan a Servo Back and Forth

Arduino

(load AnalogFirmata.pde onto Arduino)

+

Python

```
import pyduino, numpy
ard = pyduino.Arduino('/dev/ttyS0')

pin = 9

# Activate Appropriate Port
ard.digital_ports[pin >> 3].set_active(1)
ard.digital[pin].set_mode(pyduino.DIGITAL_PWM)
positions = abs(numpy.linspace(-180,180,360))

# Rotate to Different Angles
while 1:
    for pos in positions:
        ard.iterate()
        ard.digital[pin].write(pos/256)
```



Demo: servopan.py

Putting it Together: Find the Light

Python

```
import pyduino
```

```
servopin = 9  
servolev = 0.5
```

```
a = pyduino.Arduino('/dev/tty.usbserial-A6006kiD')  
a.digital_ports[servopin >> 3].set_active(1)  
a.digital[servopin].set_mode(pyduino.DIGITAL_PWM)
```

```
a.analog[2].set_active(1)  
a.analog[3].set_active(1)
```

```
def range_check(var,minval,maxval):  
    if var < minval:  
        var = minval  
    if var > maxval:  
        var = maxval  
    return var
```

```
while 1:  
    a.iterate()  
    servolev = (a.analog[2].read() - a.analog[3].read())/20.0+servolev  
    servolev = range_check(servolev,0,1)  
    a.digital[servopin].write(servolev)
```



Demo: findthelight.py

More Demos

Plotting Live Data

`plotlightlevel.py` – plot light sensor readings
`plotacc.py` – plot accelerometer readings

Accelerometer Mouse

`accmouse.py` – accelerometer-based mouse control

The End