Coding the Arduino

“Embedded Systems”

Software

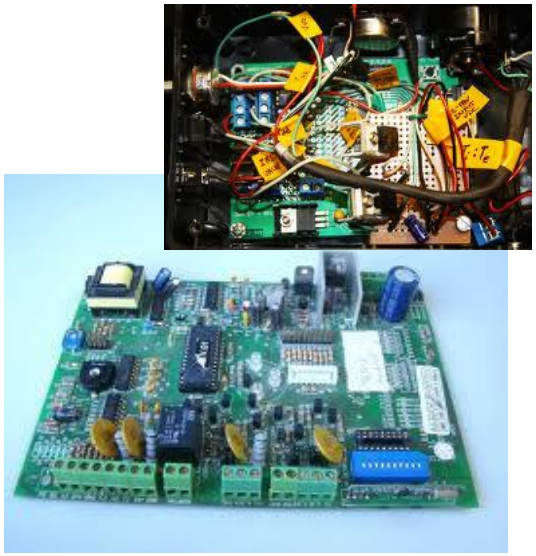
Lives in a "Virtual World"

Restart makes all OK

Easy to Modify; programmable

Expensive: i7 ~$800

Hardware



Deals with the Real World

Arduous to Modify; NOT programmable

Expensive: PCB >$100

Software

Lives in a "Virtual World"

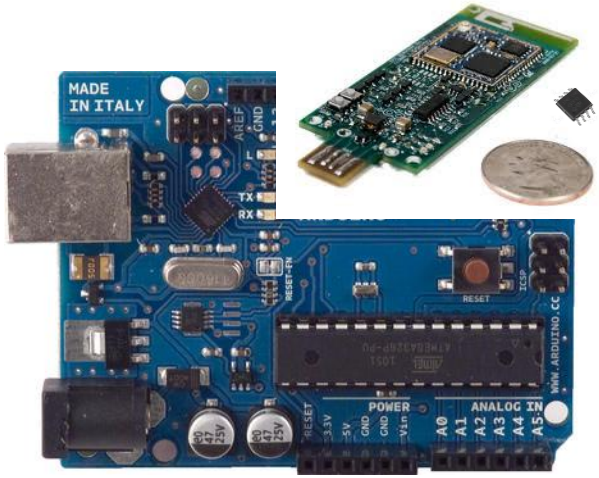
Restart makes all OK

Easy to Modify; programmable

Expensive: i7 ~$800

Firmware

"Embedded Systems"



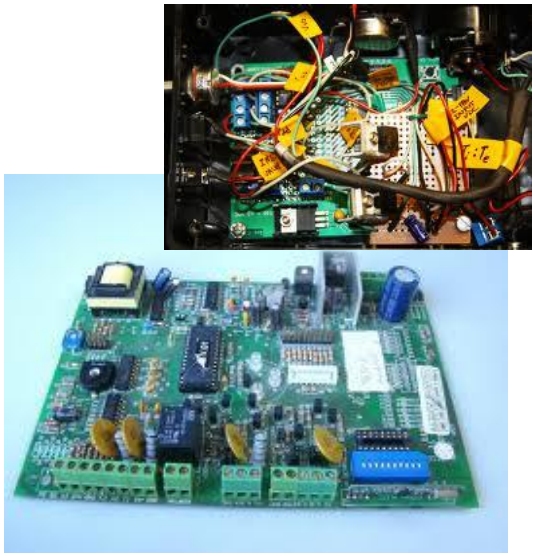
Does Both

Restart makes *Some* things OK

Easy to Modify; programmable

Inexpensive:$1-$30

Hardware



Deals with the Real World

Arduous to Modify; NOT programmable

Expensive: PCB >$100

Software

- Computers do exactly what you tell them, no more, no less
- Arduino uses the C programming language
- You can go a long ways with just a few instructions
- See the Arduino Guide (2011 web site)

On the Arduino

Programming cable

I/O pins

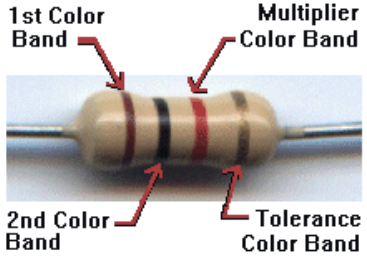
Reset

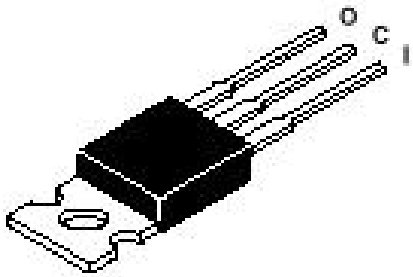
9V or 12V battery

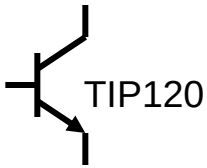
Power pins

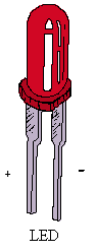
Brain

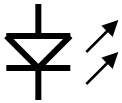
Schematic Icons: Hardware

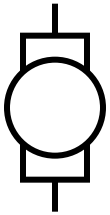


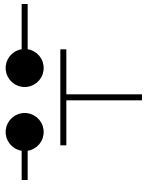




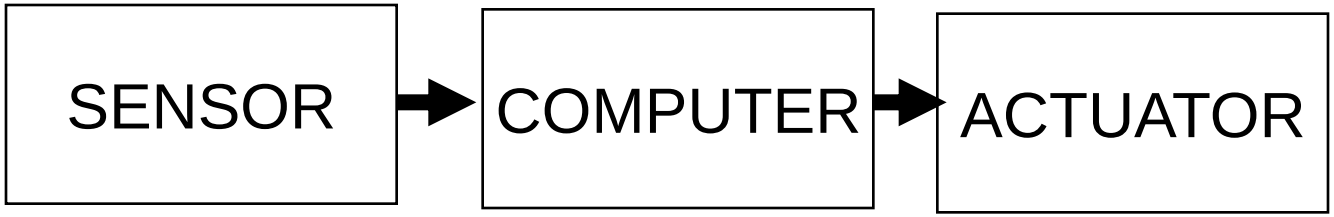








Dealing With the Outside World



Switch
Light beam
Potentiometer
Encoder
Temperature
…

Lamp
Relay
Motor
Solenoid
…

I/O Commands

DigitalWrite(*n*,HIGH); set pin *n* to +5 V

DigitalWrite(*n*,LOW); set pin *n* to 0 V

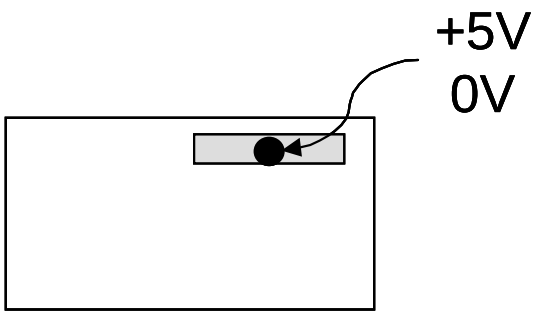
DigitalRead(*n*); read state of pin *n*

Driving Outputs

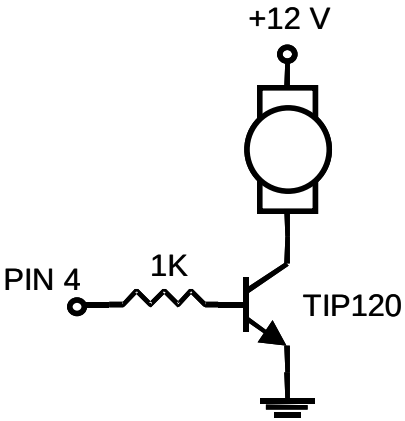
Program sets pin high/low (1/0)

```
digitalWrite(4,HIGH);
digitalWrite(4,LOW);
```

Board pin set to +5V/0V



Interface electronics use signal voltages and power supply to switch motor on/off



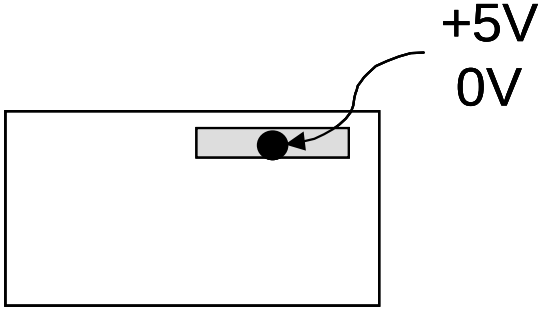
Reading Sensors

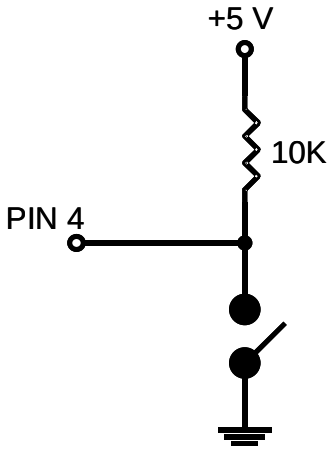
Program reads value of pins (1/0)

Board pins set to +5V/0V

Interface electronics change sensor signals into +5V/0V

```
digitalRead(4);
```





Program Structure

```
/* declare variable names here
   to be used below              */


void setup()
{
  // commands to initialize go here
}
void loop()
{
  // commands to run your machine go here
}
```

Comments

Commands

Anatomy of a Program

```
/*----------------------------
Turn on LED for ½ sec
----------------------------*/

void setup()                    // one-time actions
{
  pinMode(2,OUTPUT);            // define pin 2 as an output
  digitalWrite(2,HIGH);         // pin 2 high (LED on)
  delay(500);                   // wait 500 ms
  digitalWrite(2,LOW);          // pin 2 low (LED off)
}
void loop()                     // loop forever
{
}
```

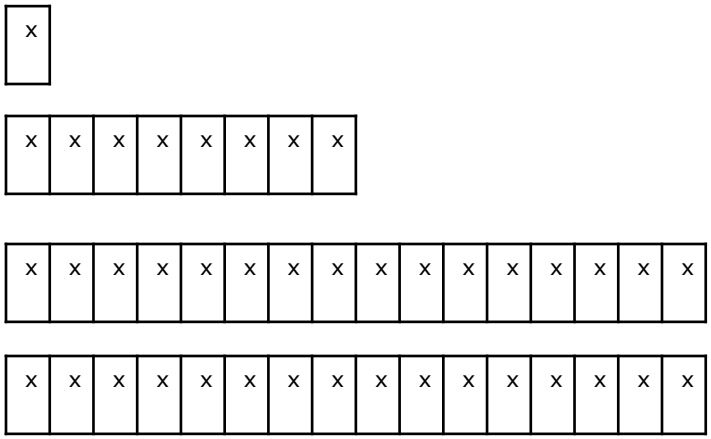
Digital Numbers

A bit is one binary digit: 0/1
A byte is 8 bits

00000011 (binary) = 3 (decimal)

11111111 (binary) = 255 (decimal)

b7							b0



Type	#bits	Number range
bit	1	0-1
byte	8	0-255
word	16	0-65,535
int	16	-32,768-32,767

Arduino Data Variables

Declare at top of program

byte i;	*0 to255*
word k;	*0 to 65,536*
int length;	*-32,768 to 32,767*
int width;	

type

name

Variable Names: Can't have white-space, use camelCase: myVariableName

Make them short but meaningful: motorSpd, redLED

Use byte variables unless expecting large numbers;
Don't mix types:byte i=266 will 'roll over' to 0

Constant Symbols

```
#define LED 2     // define the LED pin
void setup()
{
  pinMode(LED,OUTPUT);
}
void loop()
{
  digitalWrite(LED,HIGH);
  delay(500);
  digitalWrite(LED,LOW);
  delay(500);
}
```

Changeable Variables

```
#define LED 2     // define the LED pin
int myDelay = 500;
void setup()
{
  pinMode(LED,OUTPUT);
}
void loop()
{
  digitalWrite(LED,HIGH);
  delay(myDelay);
  digitalWrite(LED,LOW);
  delay(myDelay);
  myDelay = myDelay - 50;
}
```

Setting Pin Direction

```
void setup()
{
  pinMode(2,OUTPUT);
  pinMode(3,INPUT);
}
void loop() {}
```

What Does This Program Do?

```
#define LED 2    // the LED pin
byte i,j;
void setup()
{
  pinMode(LED,OUTPUT);
  for (i=0;i<4;i++) {
    flasher();
    delay(5000);
  }
}
void loop()
{}
void flasher() {
  for (j=0;j<3;j++) {
    digitalWrite(LED,HIGH);
    delay(500);
    digitalWrite(LED,LOW);
    delay(500);
  }
}
```

Printing to the Terminal

```
void setup()
{
  Serial.begin(9600);
  Serial.println("Hello World");
}
void loop() {}
```

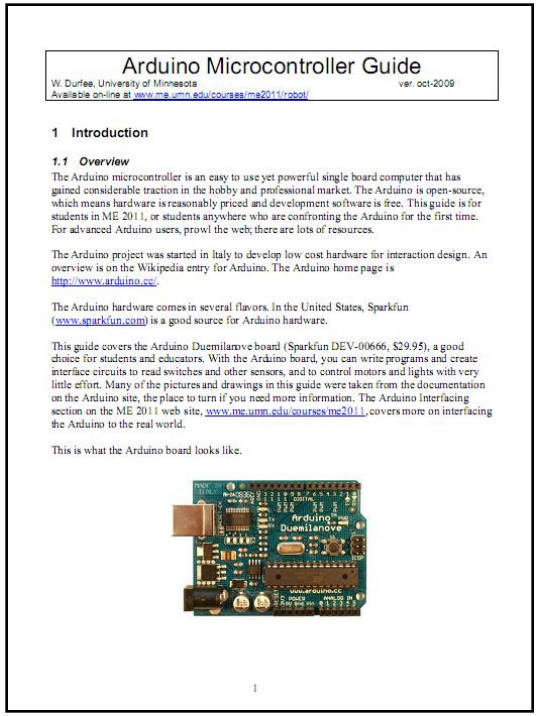
Debugging an Input

```
void setup()
{
  Serial.begin(9600);
}
void loop()
{
   // Read from digital pin 2 &
   // spit the value out to serial
   Serial.println(digitalRead(2));

   // Every 100ms or so
   delay(100);
}
```

Want More?

- "Arduino Microcontroller Guide"
- Language Reference section of Arduino site



Arduino Microcontroller Guide

W. Durfee, University of Minnesota ver. oct-2009

Available on-line at www.me.umn.edu/courses/me2011/robot/

1 Introduction

1.1 Overview

The Arduino microcontroller is an easy to use yet powerful single board computer that has gained considerable traction in the hobby and professional market. The Arduino is open-source, which means hardware is reasonably priced and development software is free. This guide is for students in ME 2011, or students anywhere who are confronting the Arduino for the first time. For advanced Arduino users, prowl the web; there are lots of resources.

The Arduino project was started in Italy to develop low cost hardware for interaction design. An overview is on the Wikipedia entry for Arduino. The Arduino home page is http://www.arduino.cc/.

The Arduino hardware comes in several flavors. In the United States, Sparkfun (www.sparkfun.com) is a good source for Arduino hardware.

This guide covers the Arduino Duemilanove board (Sparkfun DEV-00666, $29.95), a good choice for students and educators. With the Arduino board, you can write programs and create interface circuits to read switches and other sensors, and to control motors and lights with very little effort. Many of the pictures and drawings in this guide were taken from the documentation on the Arduino site, the place to turn if you need more information. The Arduino Interfacing section on the ME 2011 web site, www.me.umn.edu/courses/me2011, covers more on interfacing the Arduino to the real world.

This is what the Arduino board looks like.

1