Sources:

- Arduino Hands-on Workshop, SITI, Universidad Lusófona
- Arduino Spooky projects
- Basic electronics, University Pennsylvania
- Beginning Arduino Programming
- Getting Started With Arduino

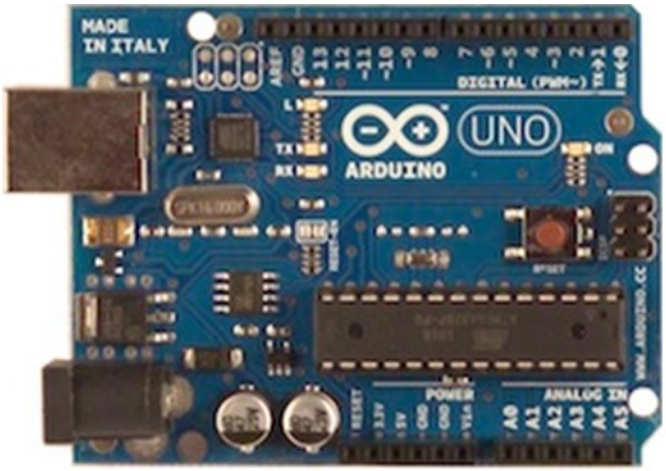
Course on Advanced Interaction Techniques

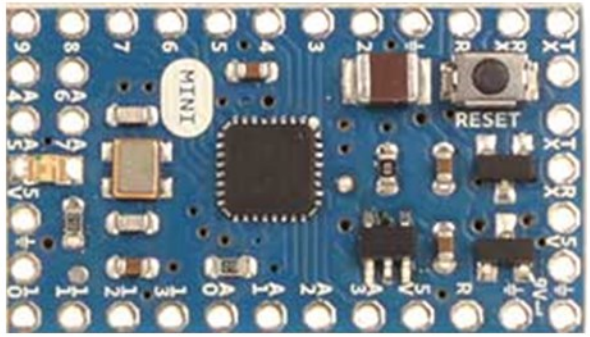
Programming ARDUINO

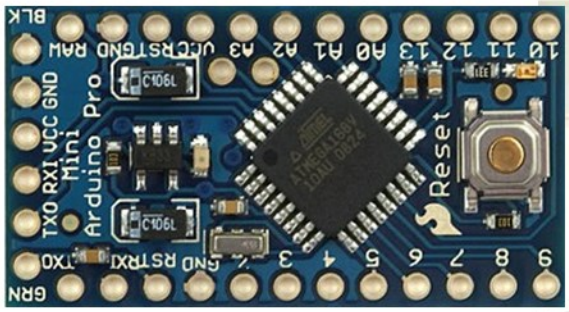
Luís Carriço

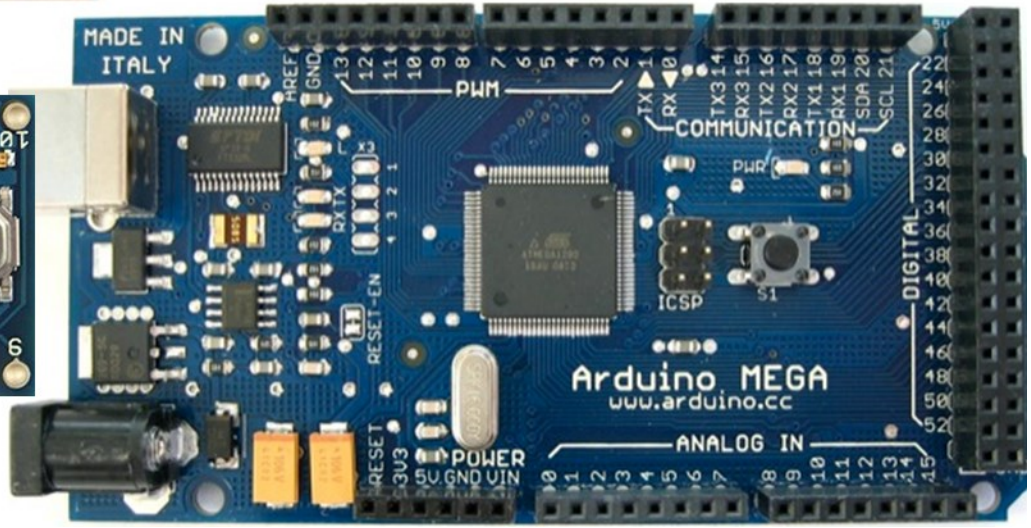
FCUL 2012/13

Arduino


MADE IN ITALY
UNO
ARDUINO
DIGITAL (PWM~)
POWER
ANALOG IN


MINI
RESET


Mini Pro
Arduino
Reset


MADE IN ITALY
PWM
COMMUNICATION
DIGITAL
ANALOG IN
POWER
ICSP
Arduino MEGA
www.arduino.cc

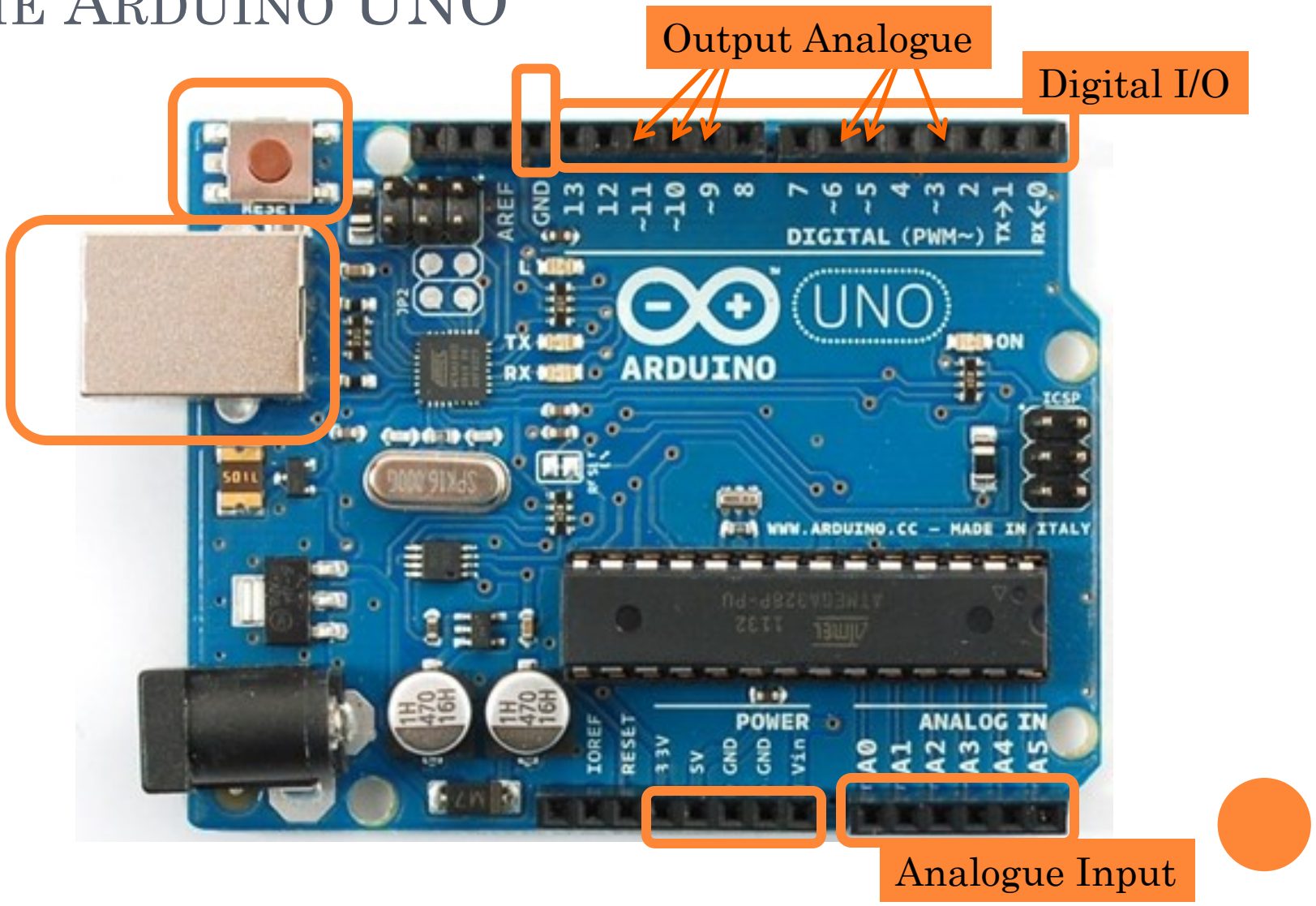
What is Arduino?

- “Arduino is an open-source physical computing platform based on a simple i/o board and a development environment that implements the Processing / Wiring language.

- Arduino can be used to develop stand-alone interactive objects or can be connected to software on your computer.“

  - www.arduino.cc, 2006

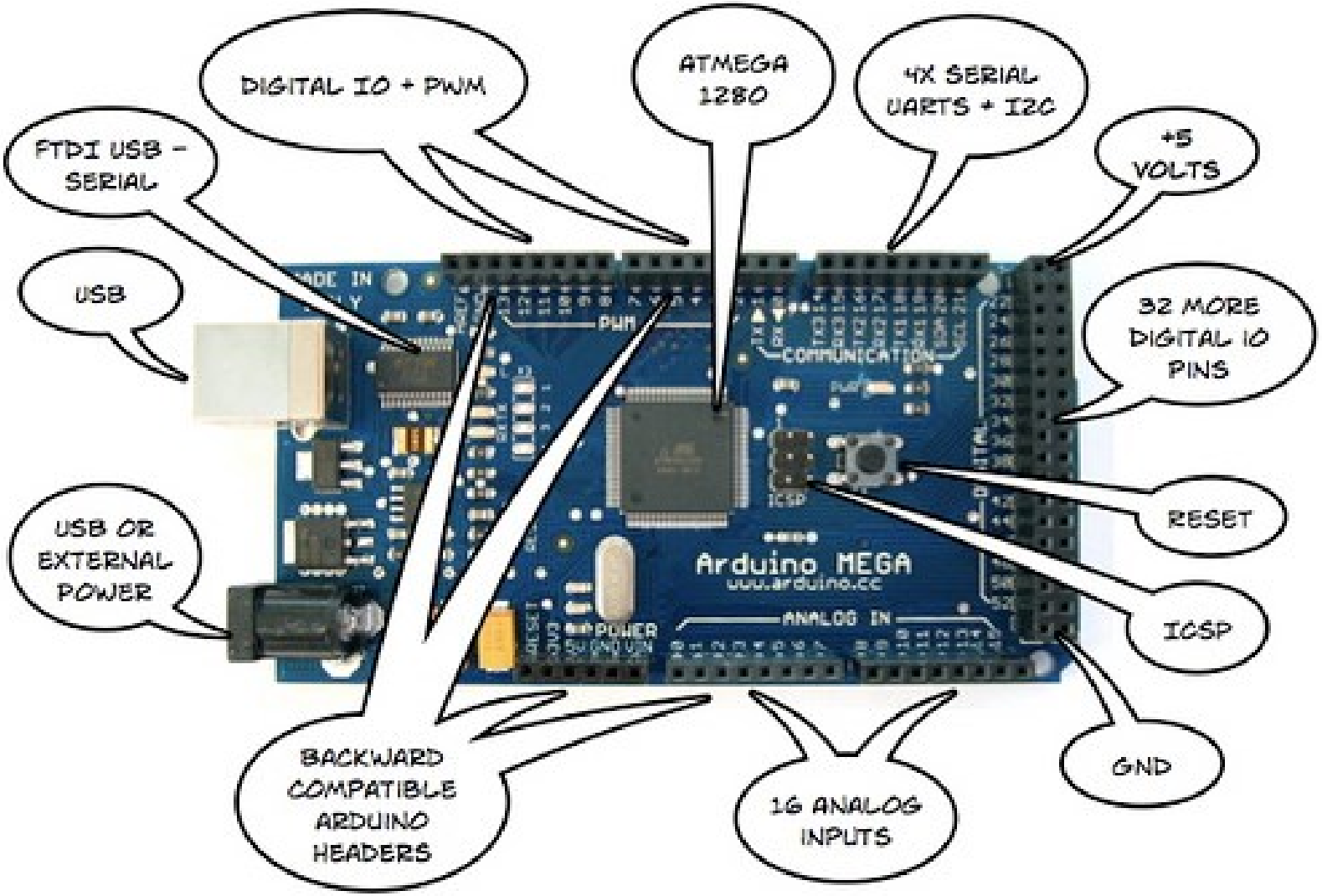
The Arduino UNO



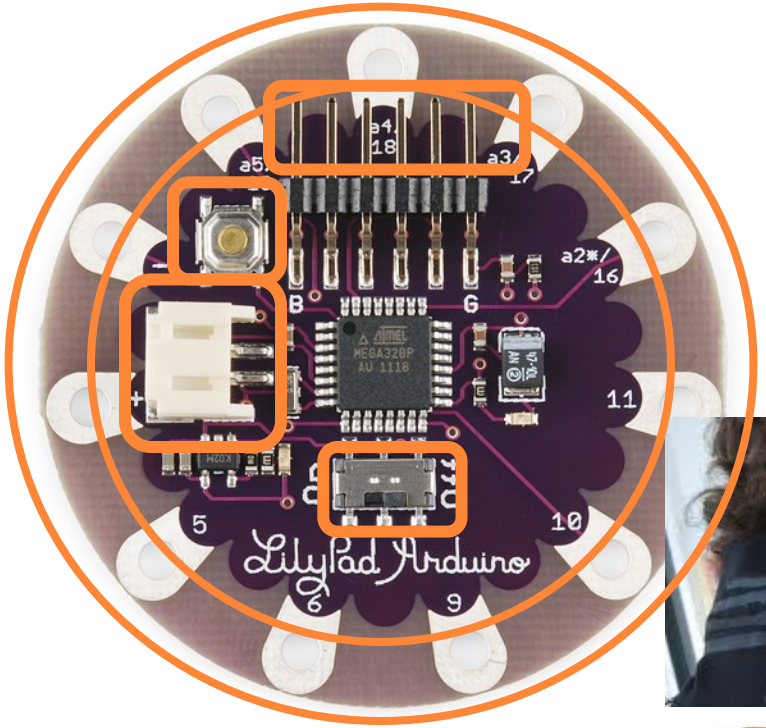
The UNO

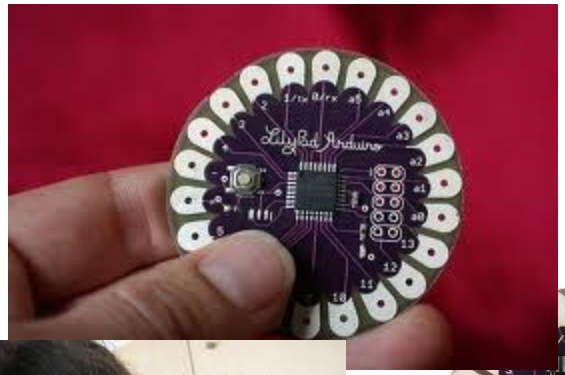
- 16 Kbytes of Flash program memory
- 1 Kbyte of RAM
- 16 MHz
- Inputs and outputs
  - 13 digital input/output pins
  - 5 analogue input pins
  - 6 analogue output pins (PWM only)

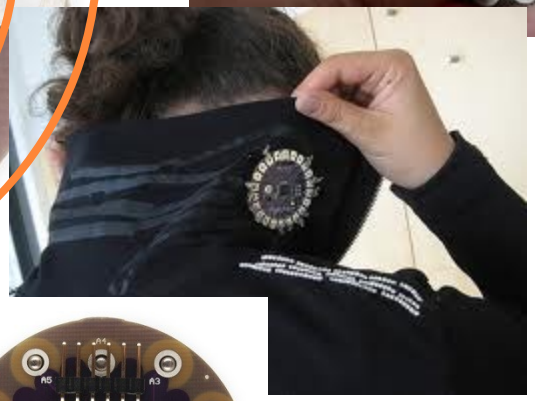
The arduino Mega

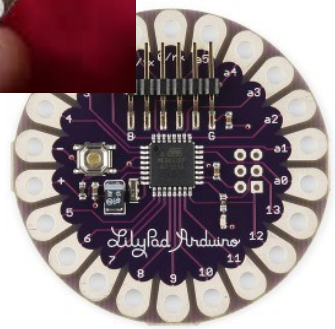


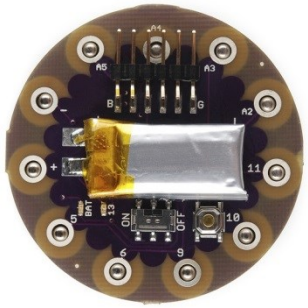
The Lylypad

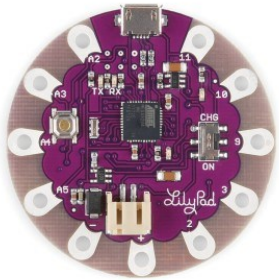








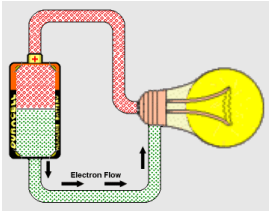




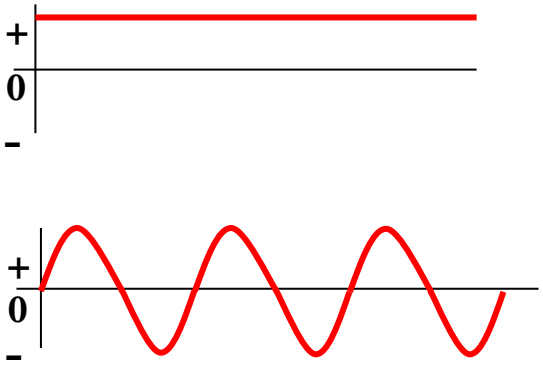
Electronics

Basic Concepts

Basic Electronics Concepts

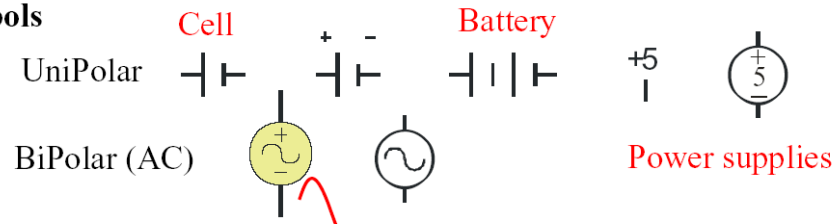


- Voltage (V)
  - electrical potential difference between two points;
  - measured in volts
- Current (I)
  - intensity of electric current;
  - measured in amps



- Voltage pushes, current flows

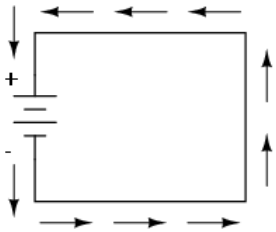
Symbols



Properties

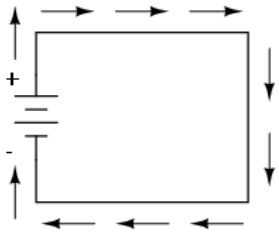
Constant Voltage, independent of the amount of current

Electron flow notation



Electric charge moves from the negative (surplus) side of the battery to the positive (deficiency) side.

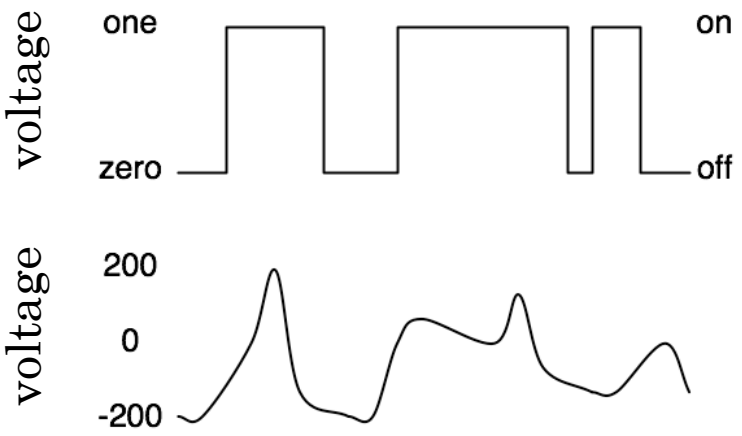
Conventional flow notation



Electric charge moves from the positive (surplus) side of the battery to the negative (deficiency) side.

Basic Electronics Concepts

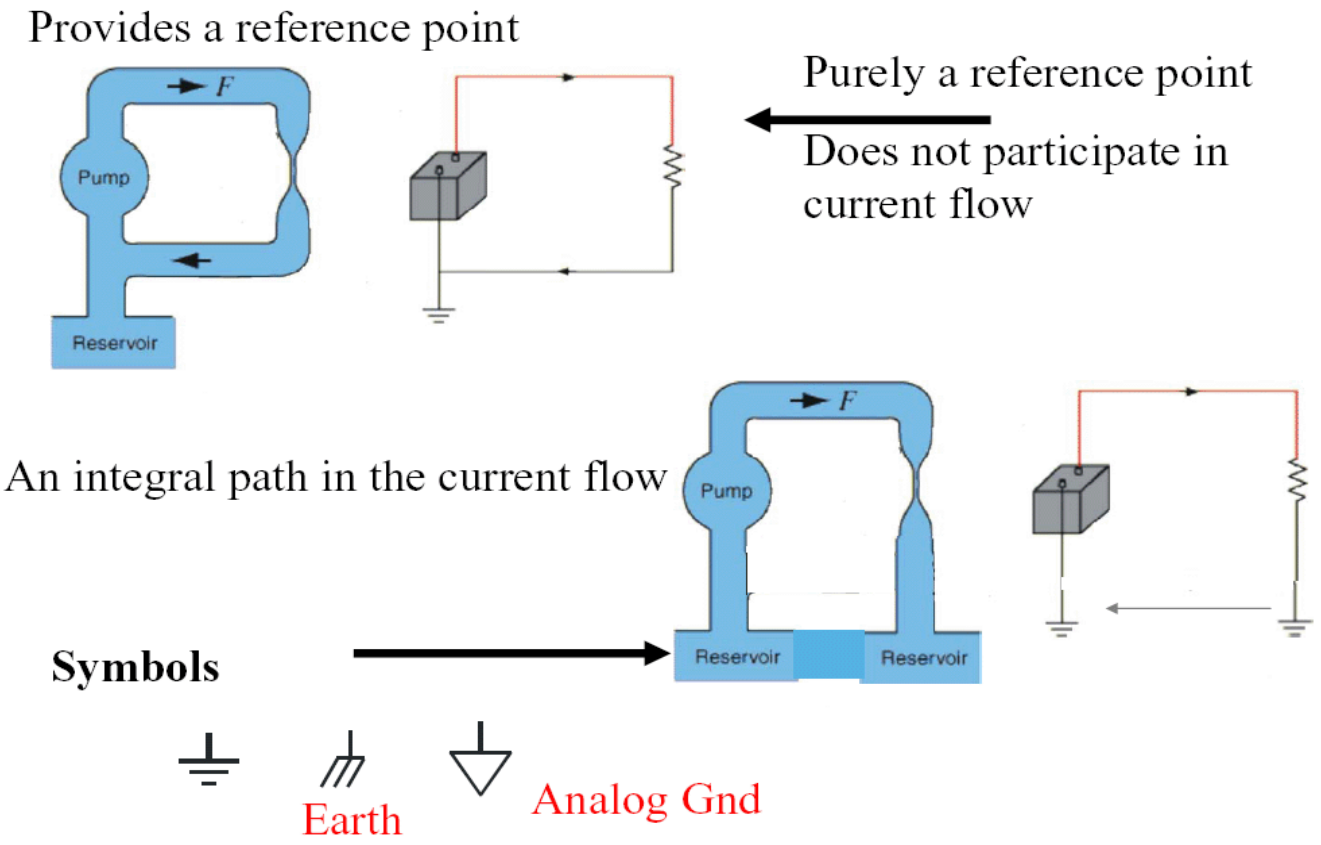
- Digital/Analogue
  - Digital is Boolean (on/off interpretation)
  - Analogue is continuous



- Computers don't really do analog,
  - so they fake it, with quantization

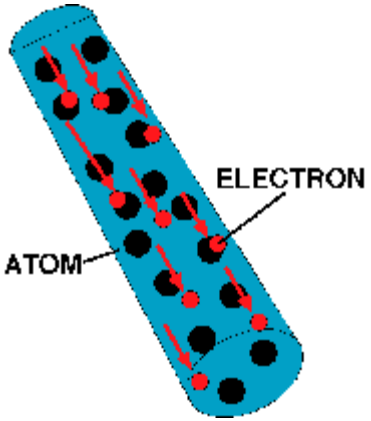
Basic Electronics Concepts

- Ground



Basic Electronics Concepts

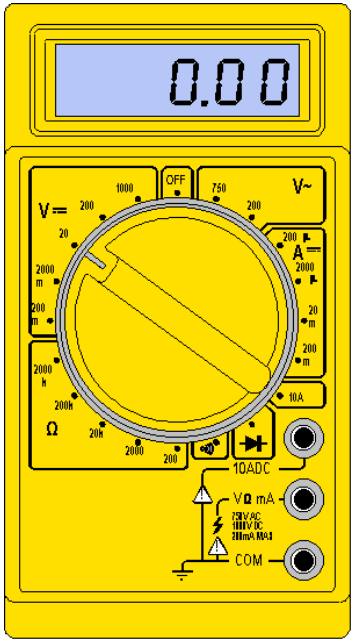
- Resistance
  - All materials have a resistance that is dependent on cross-sectional area, material type and temperature



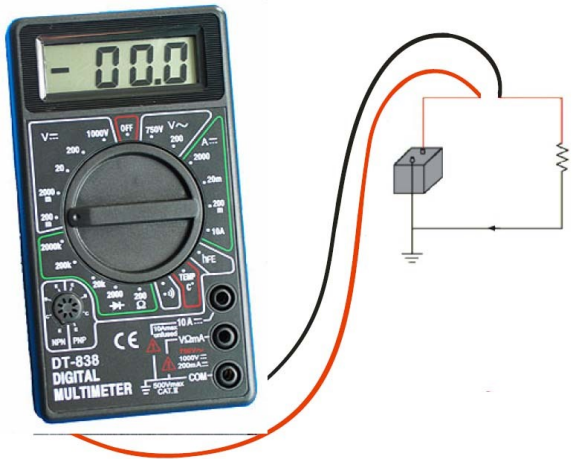
- Ohms Law

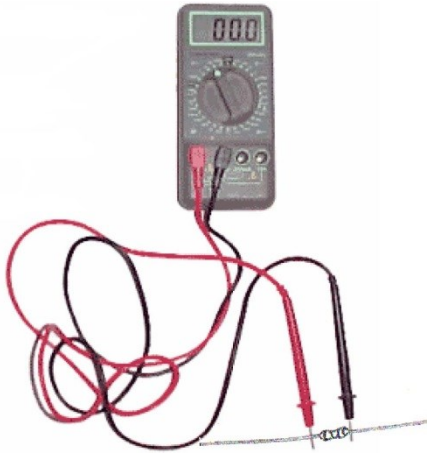
$$V = RI \qquad I = \frac{V}{R} \qquad R = \frac{V}{I}$$

The Multimeter



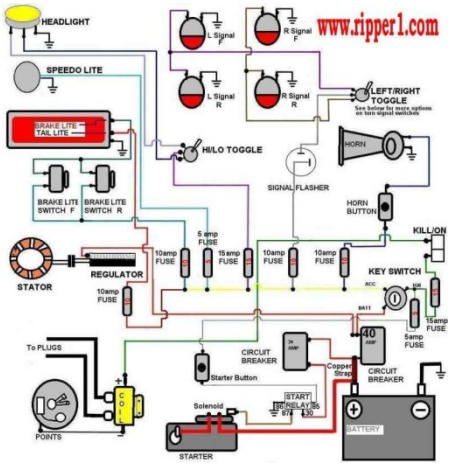
- Types of meters
  - Ammeter: measures current.
  - Voltmeter: measures the potential difference (voltage) between two points.
  - Ohmmeter measures resistance.
  - Multimeter: combines these functions and others into a single instrument



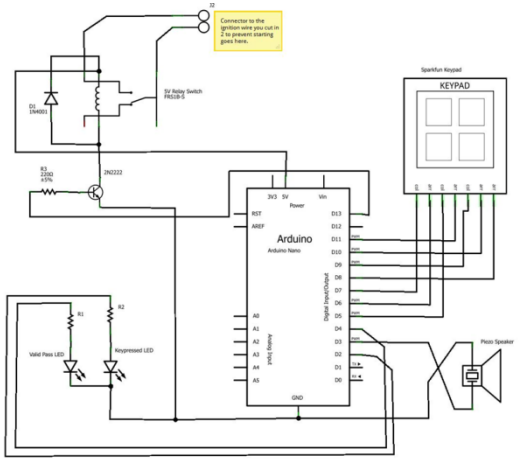


Diagrams

- Wiring drawing shows the interconnection (physical layout) of all devices and components



- Schematic drawing shows underlying logic may *not link back to physical layout*

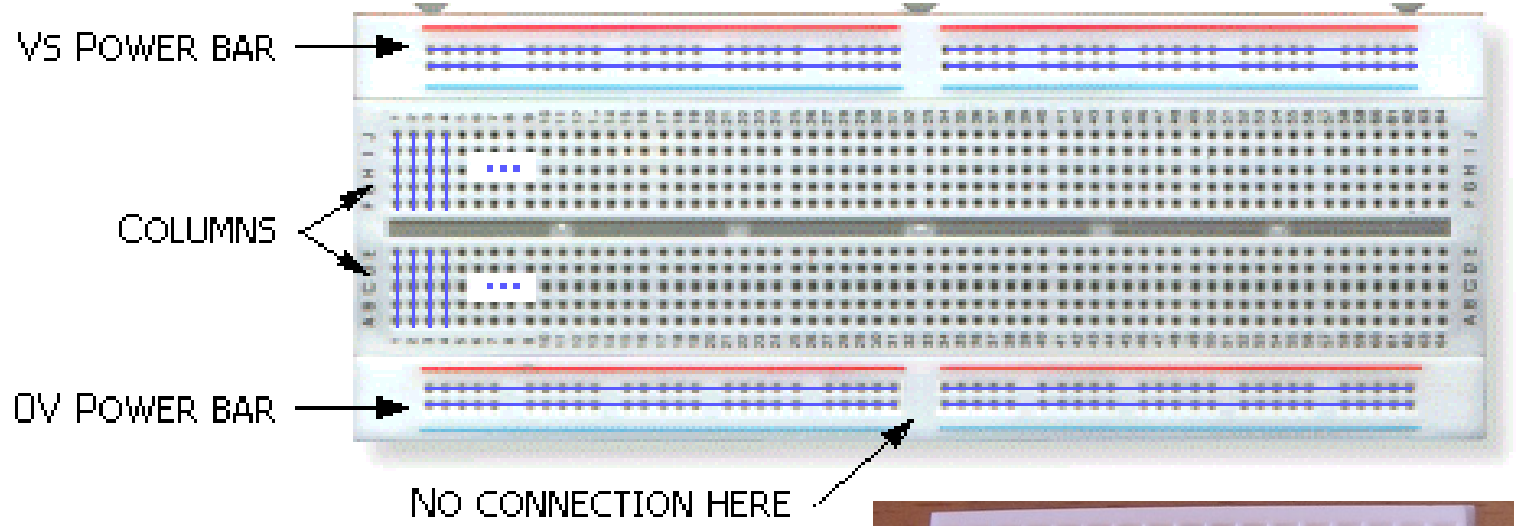


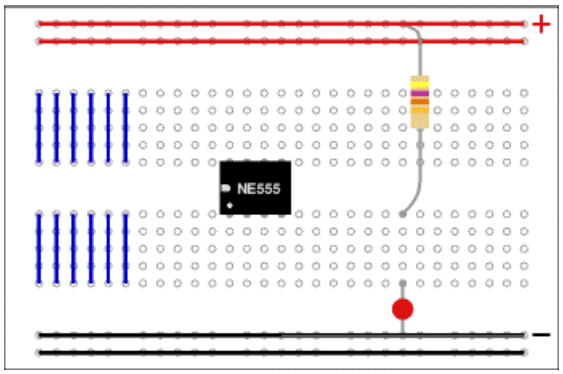
Components & Shields

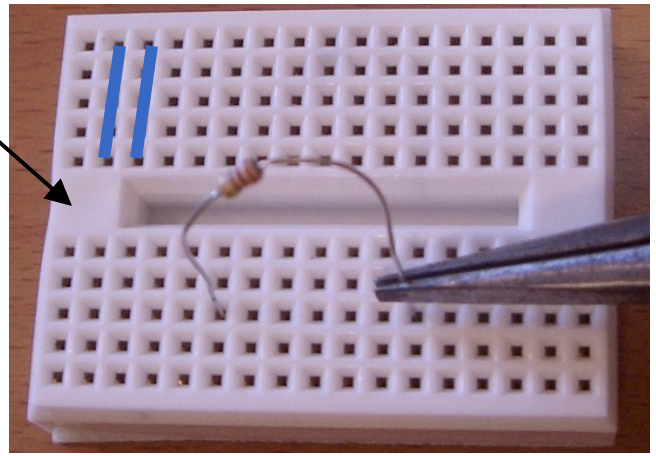
Common Components and Shields
For Arduino Projects

Components: Prototyping Board

- Used to connect components



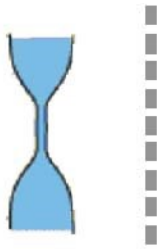




Components: the resistor

- Used to:
  - limit the intensity of current flowing on a circuit
  - transform electrical energy into heat

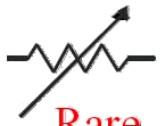
Constriction creates Resistance to water flow



Resistor creates Resistance to current flow

Symbols

3.3K Ω

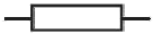


Rare

wiper

2.7M

Variable resistor



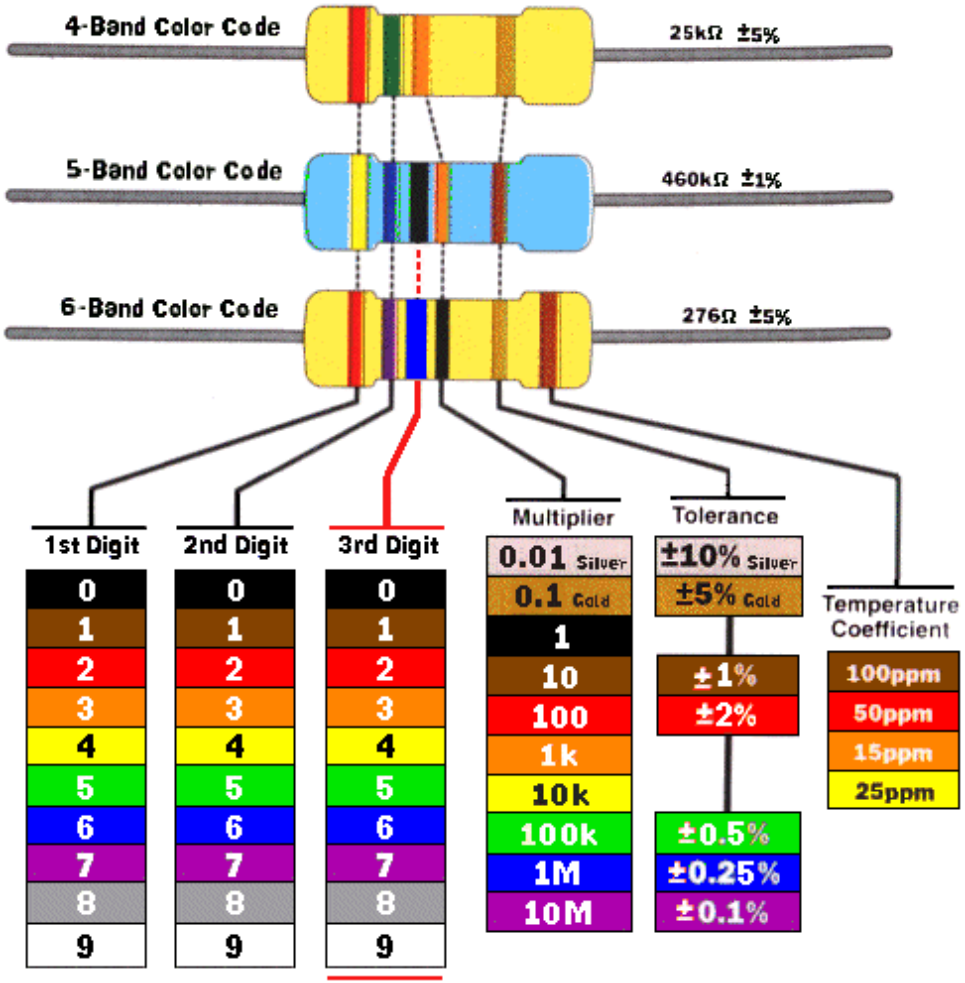
Euro

3K3

3.3k

- Re

Components: the resistor

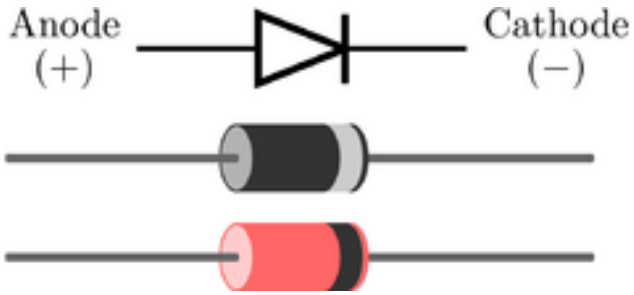


Important factors:

- tolerance of the nominal value and its stability considering the storage and operation conditions;
- the maximum power dissipated;
- the temperature coefficient;
- the maximum voltage in the terminals.

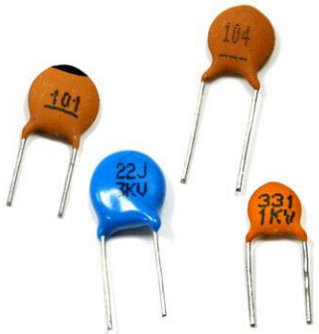
Components: the Diode

- Allows current to pass through in one direction more easily than the other
- Convert alternating current into direct current

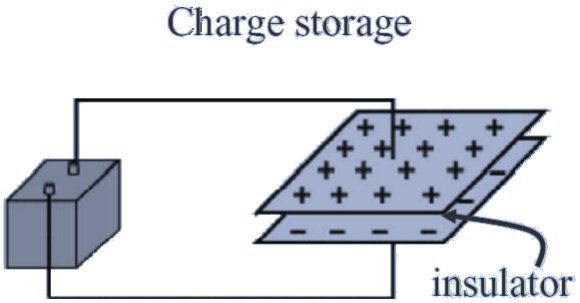


- Important factors:
  - Forward voltage drop (Vf)
  - Peak Inverse Voltage (PIV)
  - Maximum forward current
  - Leakage current

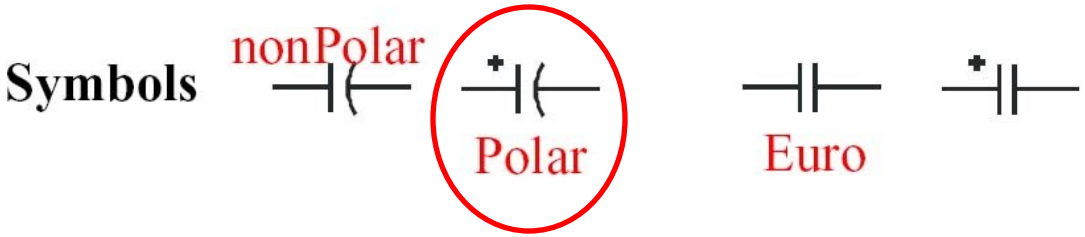
Components: the Capacitor



- Stores energy in an electric field, accumulating
  internal imbalance of electric cl

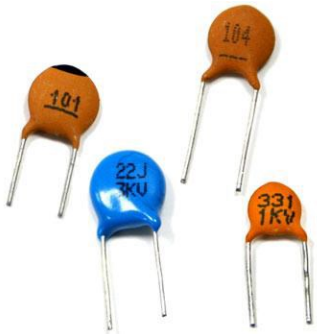


- Used to:
  - Stabilize voltage;
  - Noise filtering;
  - Separate AC DC (this is known as AC coupling).



- The storage capacity (C) is measured in farads (F).

Components: the Capacitor

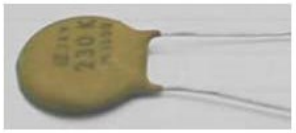


- Types
  - ceramics (low values up to about 1 mF);
  - polystyrene (usually in the range of picofarads);
  - polyester (about 1 nF to 10 mF);
  - polypropylene (low loss, high voltage, resistant to breakdown);
  - tantalum (compact, low-voltage device, up to about 100 mF);
  - electrolyte (high power, compact but lossy, in the range of 1 mF to 1000 mF);

Mylar

Monolythic Cermamic

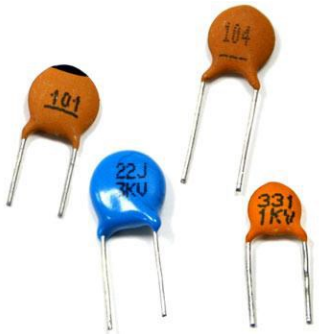
Tantalum



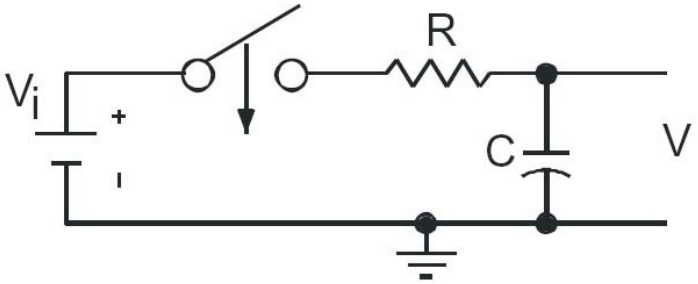
Ceramic

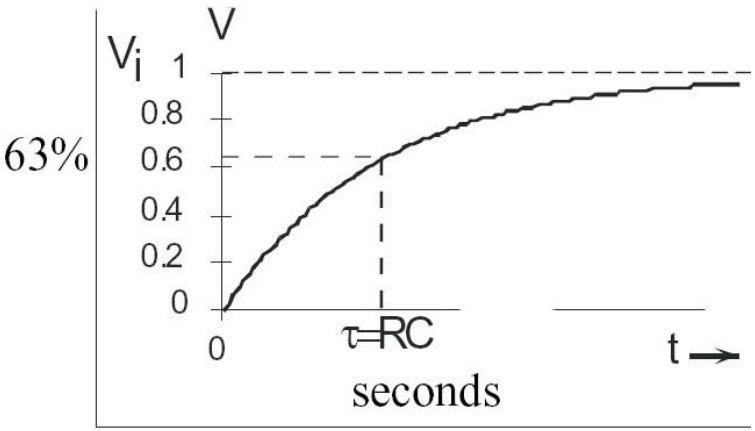
Electrolytic

Components: the Capacitor

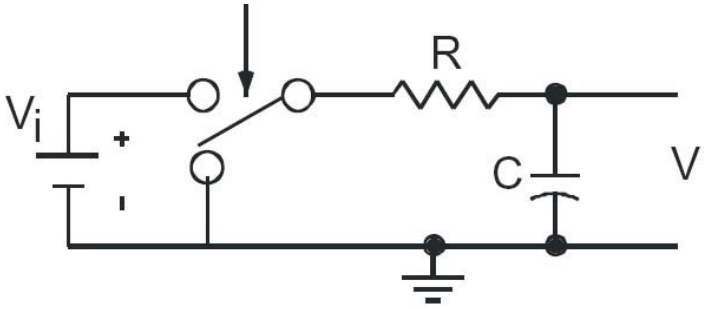


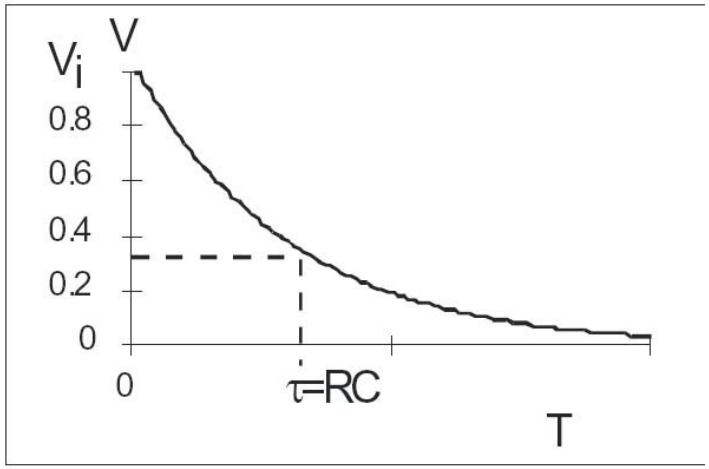
$V = V_i - V_i e^{-t / RC}$





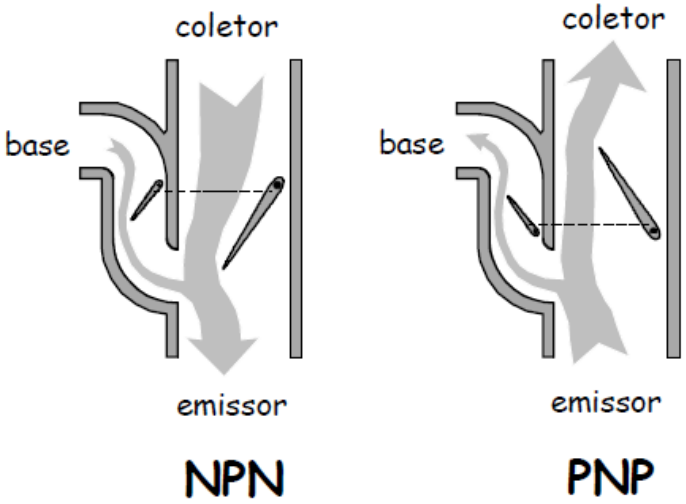
$V = V_i e^{-t / RC}$

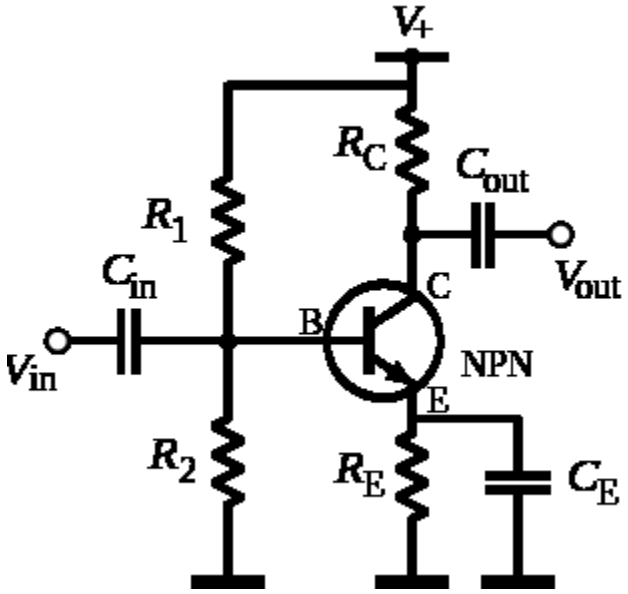




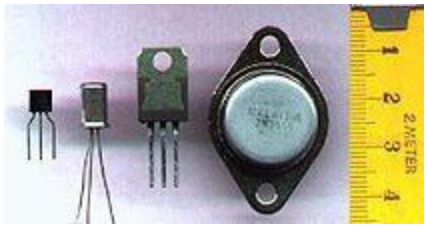
Components: the Transistor

- Used as:
  - amplifiers and switches for electrical signals.

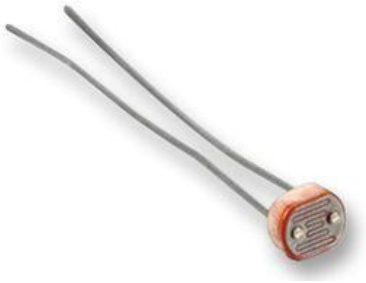




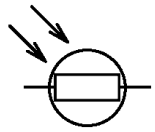
- Should find which pin is which!

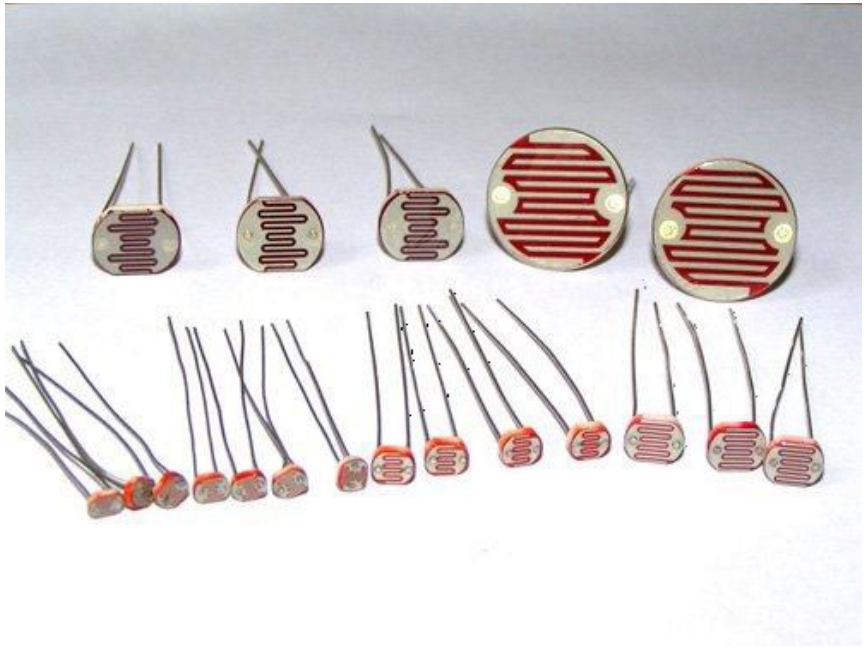


Components: the LDR



- Light Dependent Resistor (LDR)
  - is a resistor whose resistance decreases with increasing incident light intensity. It can also be referred to as a photoconductor.

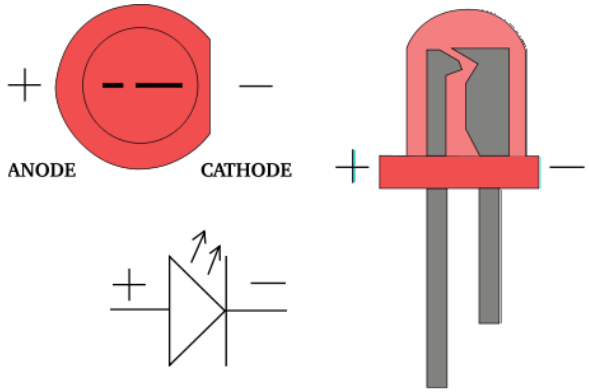


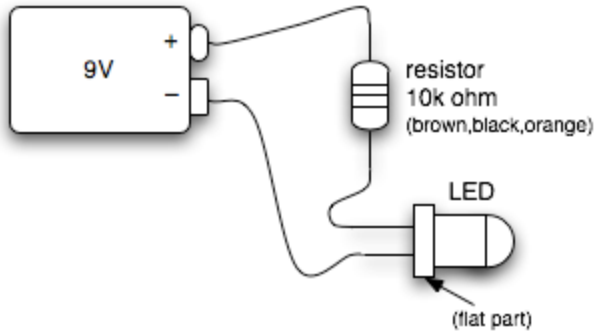


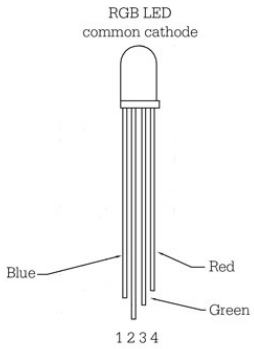
- http://en.wikipedia.org/wiki/Light_Dependent_Resistor

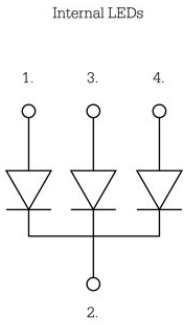
Components: the LED

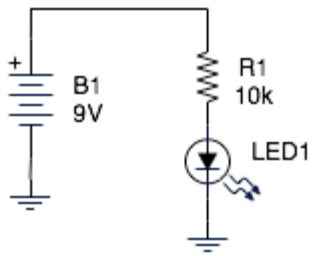
- LED = Light-Emitting Diode
  - it is a diode
  - Emits light









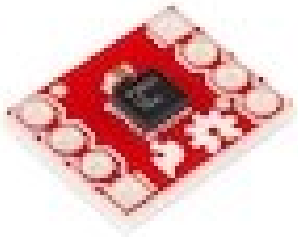


- Important factors:
  - Needs a “current limiting” resistor, or burns out

Components: sensors

Vibration

Tilt



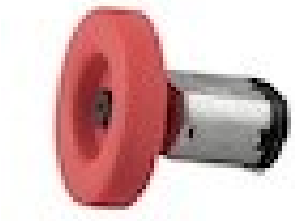
Accelerometer

Temperature

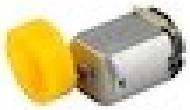
Components: Actuators & Controllers

Vibration

Buttons



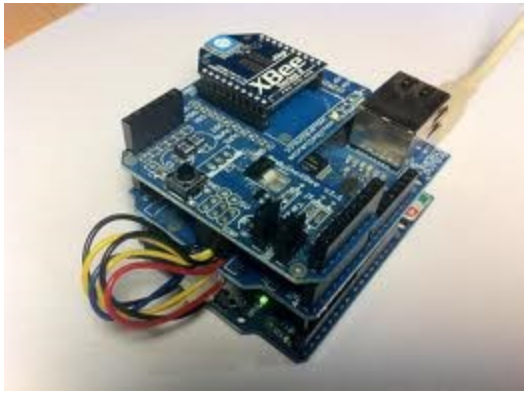
Buzzer



Motor

Potenciometer

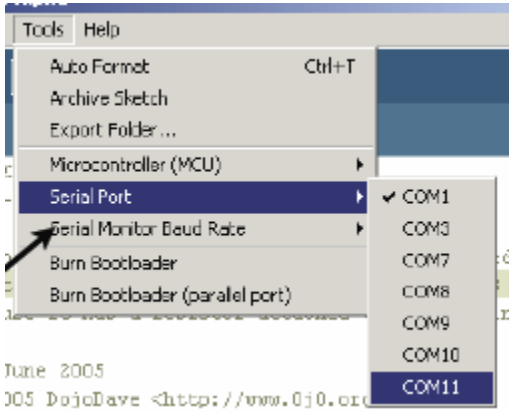
Shields

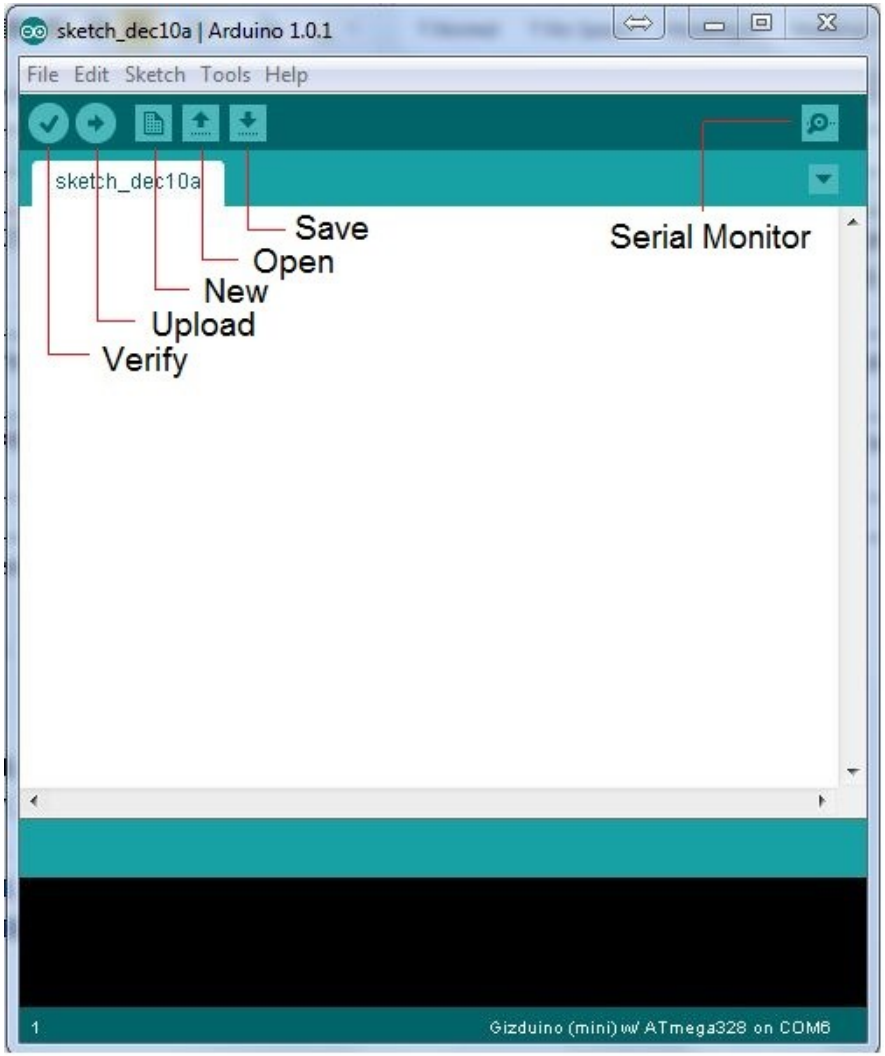


Programming

Programming the Arduino

The arduino Platform





Skeching the Arduino

- Write the ``*sketch*'' - program that runs on the board
- Check it
- Upload it
- It's running – may reset

```
void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);   // set the LED on
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // set the LED off
  delay(1000);              // wait for a second
}
```

Na Arduino Sketch

- Declare at the top
  - variables and constants
- Initialize
  - setup()
  - run once at beginning, set pins
- Running
  - loop()
  - run repeatedly

Na Arduino Sketch - Declare

```
// led connected to control pin 13
int ledPin = 13;


// setup sensor 'aSensor' on analog pin 0
int aSensor = 0;


// use this to hold the state of a pin
int statePin = LOW;
```

- Constants
  - LOW, HIGH
  - INPUT, OUTPUT
  - …
- Types
  - boolean, byte, int unsigned, … float, …
  - Arrays [] with {}
- ….
- It’s processing
- It’s a kind of C

Na Arduino Sketch - Setup

```
pinMode(ledPin, Output);
// set the pin `ledPin' as an
output


serial.Begin(9600);
// talk to the computer at
// 9600 baud rate
```

- There's not much more than this

Na Arduino Sketch – loop functions

Input and output functions

```
digitalWrite(8,HIGH);
// turns on digital pin 8

val = digitalRead(7);
// reads pin 7 into val

val = analogRead(0);
// reads analog input 0 into val (0..1024)

analogWrite(9,128);
// Dim an LED on pin 9 to 50%

time = pulsein(7,HIGH);
// measures the time the next
// pulse stays high
```

Time functions

```
duration = millis()-lastTime;
// computes time elapsed since
"lastTime“

delay(500);
// stops the program for half a second

delayMicroseconds(1000);
// waits for 1 millisecond

/
```

Na Arduino Sketch – loop functions

Math functions

```
val = min(10,20);
// val is now 10

val = max(10,20);
// val is now 20

val = abs(-5);
// val is now 5

val = constrain(analogRead(0), 0, 255);
// reject values bigger than 255

val = map(analogRead(0),0,1023,100, 200);
// maps the value of analog 0
//to a value between 100 and 200
```

(Cont.)

```
double x = pow(y, 32);
// sets x to y raised to the 32nd power
double a = sqrt(1138);
// approximately 33.73425674438

double sine = sin(2);
// approximately 0.90929737091
// aslo for cosine and tangent

randomSeed(analogRead(5));
// randomize using noise from pin 5

long randnum = random(0, 100);
// a number between 0 and 99
long randnum = random(11);
// a number between 0 and 10
```

Na Arduino Sketch – loop functions

Serial communication functions

```
Serial.begin(9600);
// usually no more than 115,200 bps.

Serial.print(75);
// Prints "75"
Serial.print(75, DEC);
// The same as above.
Serial.print(75, HEX);
// "4B" (75 in hexadecimal)
Serial.print(75, OCT);
// "113" (75 in octal)
Serial.print(75, BIN);
// "1001011" (75 in binary)
```

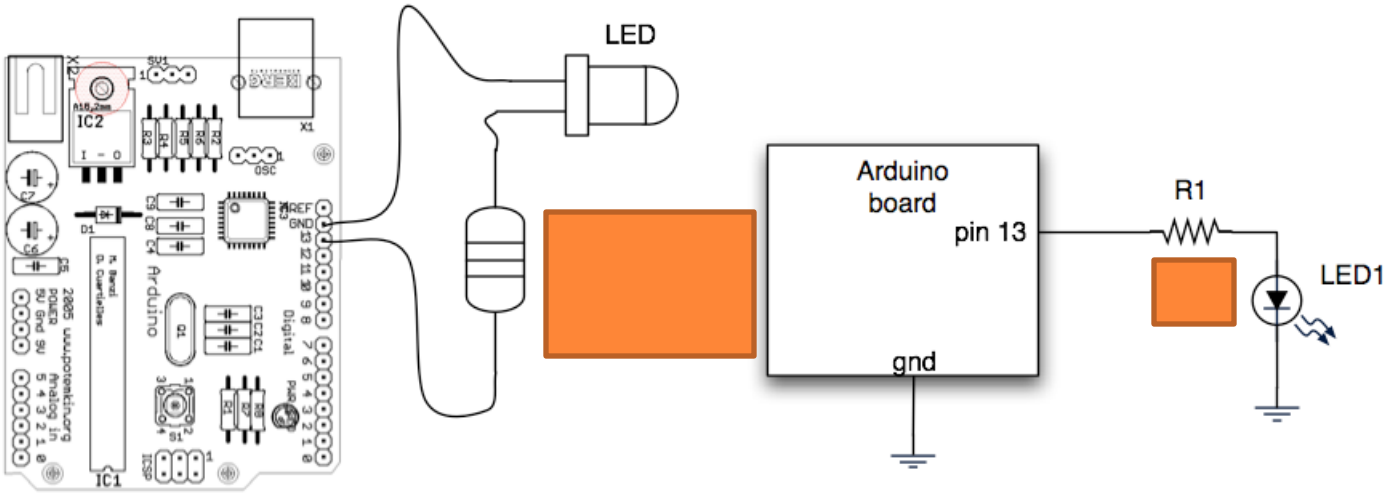
(Cont.)

```
Serial.println(75);
// Prints "75\r\n"
Serial.println(75, DEC);
// The same as above.
Serial.println(75, HEX);
// "4B\r\n"
Serial.println(75, OCT);
// "113\r\n"
Serial.println(75, BIN);
// "1001011\r\n"

int count = Serial.available(); // bytes ready
int data = Serial.read(); // get one byte
Serial.flush(); // cleans the input buffer
```

// Example 01 : Blinking LED

```
const int LED = 13;                 // LED connected to
                                    // digital pin 13


void setup()
{
    pinMode(LED, OUTPUT);           // sets the digital
                                    // pin as output
}
void loop()
{
    digitalWrite(LED, HIGH);        // turns the LED on
    delay(1000);                    // waits for a second
    digitalWrite(LED, LOW);         // turns the LED off
    delay(1000);                    // waits for a second
}
```

Assignment 1: Digital Output



Caution with Specs

ARDUINO specs

- I/O
  - they operate at 5 volts
  - provide or receive a maximum of 40 mA
- DC Current
  - 3.3V Pin max 50 mA
- +5 V is out
- All at
  - http://arduino.cc/en/Main/arduinoBoardUno

LED specs

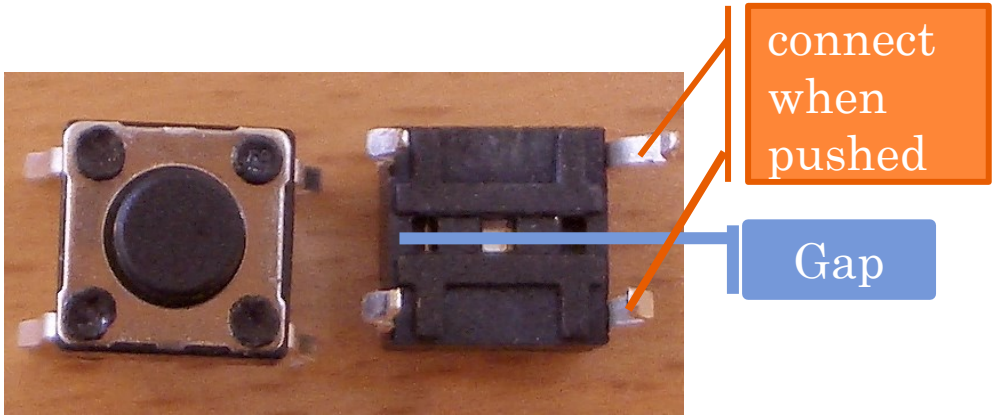
Colour	I_F max.	V_F typ.	V_F max.	V_R max.	Luminous intensity
Red	**30mA**	**1.7V**	2.1V	5V	5mcd @ 10mA
Bright red	**30mA**	**2.0V**	2.5V	5V	80mcd @ 10mA
Yellow	**30mA**	**2.1V**	2.5V	5V	32mcd @ 10mA
Green	**25mA**	**2.2V**	2.5V	5V	32mcd @ 10mA
Blue	**30mA**	**4.5V**	5.5V	5V	60mcd @ 20mA
Red	**30mA**	**1.85V**	2.5V	5V	500mcd @ 20mA
Red	**30mA**	**1.7V**	2.0V	5V	5mcd @ 2mA

Assignment 1: The code

```
int ledPin = 13;                // LED connected to digital pin 13

void setup()
{
  pinMode(ledPin, OUTPUT);      // sets the digital pin as output
}

void loop()
{
  digitalWrite(ledPin, HIGH);   // sets the LED on
  delay(1000);                  // waits for a second
  digitalWrite(ledPin, LOW);    // sets the LED off
  delay(1000);                  // waits for a second
}
```

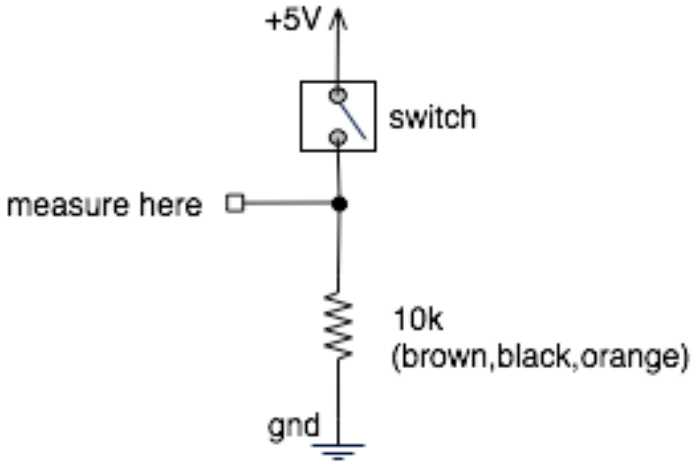
Assignment 2: Switches

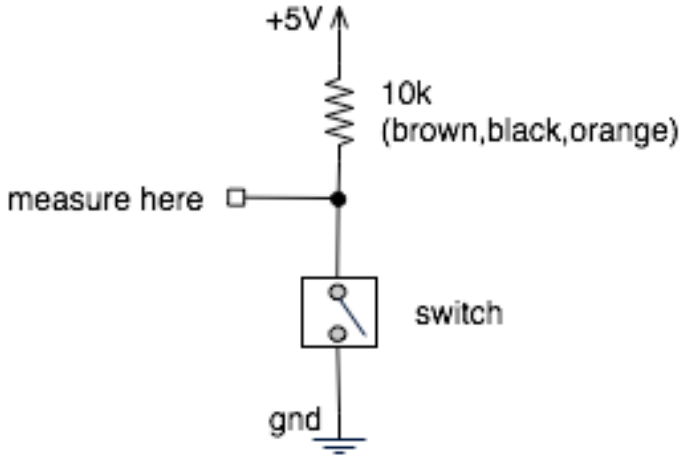
- Switches make or break a connection
- Arduino needs a voltage
  - Specifically, a "HIGH" (5 volts)
  - or a "LOW" (0 volts)



- Pressing the button, "closes the gap"
- How do you go from make/break to high/low?

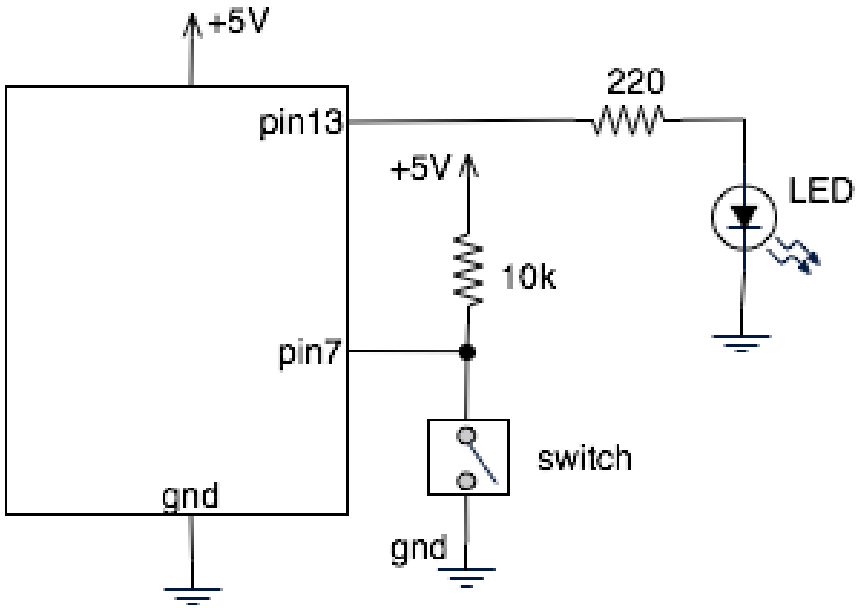
Pull Down/Up Resistors





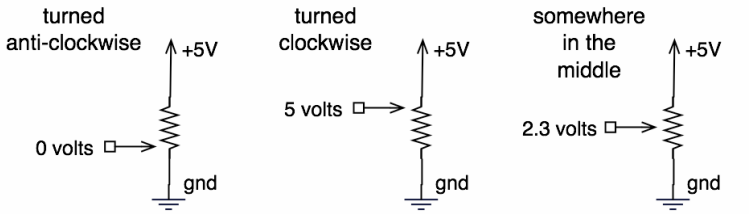
Assignment 2: Digital Input & Output

- Add switch circuit to any digital input (except pin 13)
- For output, use either existing pin 13 LED or wire up your own



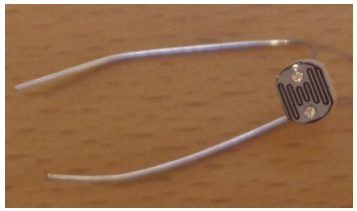
Assignment 3: Analogue Input

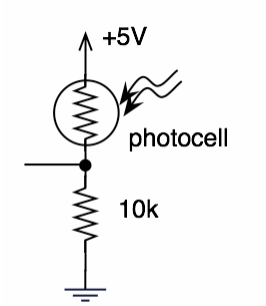
The potenciometer



The Photocell

- A variable resistor
- Brighter light == lower resistance
- Photocells you have range approx. 0-10k



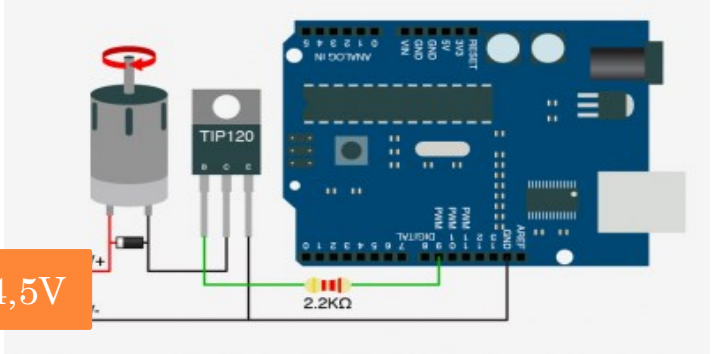


analogRead ()
- Resolution is 10-bit (1024 states)
- In other words, 5/1024 = 4.8 mV smallest voltage change you can measure

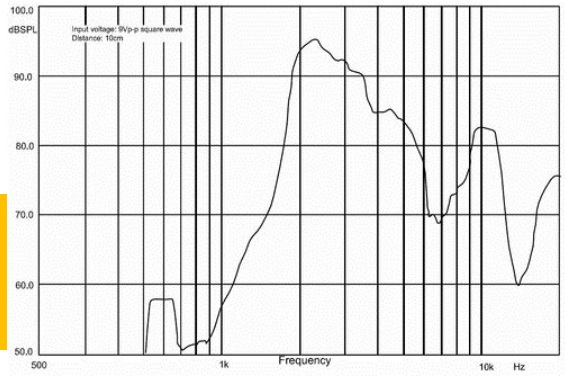
Assignment 3: Analogue Stuff

- Input
  - Use the potentiometer (e.g. to regulate colour)
  - Use the light sensor (e.g. to regulate intensity)

- Output
  - Use the RBG LED
  - Use a motor (specs!!!)
    - Motor INM-0411
      - 1,5V to 4,5V
      - Load current: 0.35A
    - Arduino UNO
      - I/O 40mA
    - http://bildr.org/2011/03/high-power-control-with-arduino-and-tip120/
  - Use a buzzer (specs!!!)
    - 1 - 30V
    - low current draw (5 mA at 9V)



1,5 - 4,5V



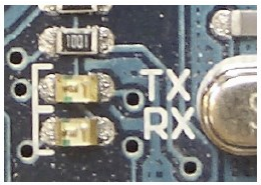
analogWrite()
- Resolution is 8-bit (256 states)
- Avalailable in specific Pins

Assignment 4: Communication Basics

- Talking to other devices uses the "Serial" commands
  - Serial.begin() – prepare to use serial
  - Serial.print() – send data to computer
  - Serial.read() – read data from computer

```
void setup() {
      Serial.begin(9600);
}

void loop()
{
  Serial.print("Light the LED [y/n]? ");
  while (Serial.available() == 0);

  data = Serial.read();
  if (data == 'y')
    digitalWrite(led, HIGH);
  else
    digitalWrite(led, LOW);

  tosend[0] = data;
  Serial.println(tosend);
}
```

- TX/RX LEDs
  - TX – sending to PC
  - RX – receiving from PC
  - Used when programming or communicating



Assignment 5: Communication C#

```
using System;
using System.IO.Ports;

public partial class MainWindow : Window {
    SerialPort arduino;

    public MainWindow() {
        InitializeComponent();
        _arduino = new SerialPort("COM5", 9600);
        _arduino.DataReceived += _arduino_DataReceived;
        try {
            _arduino.Open();
        } catch {
            MessageBox.Show("Could not connect");
        }
    }
    void _arduino_DataReceived(object sender, SerialDataReceivedEventArgs e  {
        try {
            string data = _arduino.ReadLine();
            this.Dispatcher.Invoke(new Action(() => { this._label.Content = data; }));
        } catch (Exception ex) {
            MessageBox.Show("Could not read data" + ex.StackTrace);
        }
    }
     private void myAppChecked(object sender, RoutedEventArgs e)
    {
        CheckBox chk = e.Source as CheckBox;
        if ((bool)chk.IsChecked)
            _arduino.Write("y");
        else
            _arduino.Write("n");
    }
}
```

ARDUINO

```
void loop()
{
  Serial.print("Light the LED [y/n]? ");
  Serial.println(ledstatus);
  while (Serial.available() == 0);

  data = Serial.read();
  if (data == 'y')
    digitalWrite(led, HIGH);
  else
    digitalWrite(led, LOW);

  ledstatus[0] = data;
}
```